

Encyclopaedia

Author and Editor

Elzein Salah Elzein

IEEE Member

- Papers in Electronics

IEEE Publisher, UK



Article

Selective Grasping for Complex-Shaped Parts Using Topological Skeleton Extraction

Andrea Pennisi ¹, Monica Sileo ², Domenico Daniele Bloisi ¹ and Francesco Pierri ^{2,*}

¹ Department of International Humanities and Social Sciences, UNINT International University of Rome, 00147 Rome, Italy; andrea.pennisi@gmail.com (A.P.); domenico.bloisi@unint.eu (D.D.B.)

² School of Engineering, University of Basilicata, 85100 Potenza, Italy; monica.sileo@unibas.it

* Correspondence: francesco.pierri@unibas.it; Tel.: +39-0971-205020

Abstract: To enhance the autonomy and flexibility of robotic systems, a crucial role is played by the capacity to perceive and grasp objects. More in detail, robot manipulators must detect the presence of the objects within their workspace, identify the grasping point, and compute a trajectory for approaching the objects with a pose of the end-effector suitable for performing the task. These can be challenging tasks in the presence of complex geometries, where multiple grasping-point candidates can be detected. In this paper, we present a novel approach for dealing with complex-shaped automotive parts consisting of a deep-learning-based method for topological skeleton extraction and an active grasping pose selection mechanism. In particular, we use a modified version of the well-known Lightweight OpenPose algorithm to estimate the topological skeleton of real-world automotive parts. The estimated skeleton is used to select the best grasping pose for the object at hand. Our approach is designed to be more computationally efficient with respect to other existing grasping pose detection methods. Quantitative experiments conducted with a 7 DoF manipulator on different real-world automotive components demonstrate the effectiveness of the proposed approach with a success rate of 87.04%.

Keywords: computer vision for manufacturing; deep learning in grasping and manipulation; visual learning



Citation: Pennisi, A.; Sileo, M.; Bloisi, D.D.; Pierri, F. Selective Grasping for Complex-Shaped Parts Using Topological Skeleton Extraction. *Electronics* **2024**, *13*, 3021. <https://doi.org/10.3390/electronics13153021>

Academic Editors: Krzysztof Okarma and Piotr Lech

Received: 4 June 2024

Revised: 23 July 2024

Accepted: 29 July 2024

Published: 31 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Components with complex geometrical shapes are largely used in the manufacturing industry, e.g., in the automotive sector. Using robots to handle complex-shaped parts is still a challenging task due to perception, planning, and reasoning problems. In particular, uncertainties in the position of the object to grasp and perception noise due to reflective materials are common challenges in industrial scenarios.

Grasping objects with complex geometries can be roughly classified into model-based methods that rely on pre-existing 3D models and learning-based techniques that employ machine learning to predict grasping points. Both the approaches need to perceive the external environment using vision-based algorithms, based on cameras and point clouds for object detection, segmentation, and pose estimation, as in [1,2], or tactile-based strategies, as in [3], requiring sensors for force measurement, haptic feedback, and slip detection. Hybrid approaches combine multiple methods for robustness, including multi-sensor fusion and active perception.

In this paper, we present a complete pipeline for handling complex-shaped automotive parts using a 7 DoF robot manipulator. In particular, we adopt a deep learning-based approach to design a multi-object detector aimed to extract the topological skeleton belonging to the part to grasp to precisely estimate its pose. After estimating the pose, a selection of the best grasping pose is carried out to increase the chance of grasping the object successfully.

The contribution of this work is three-fold.

1. The skeleton extraction process provides a representation of the object pose in 3D space. Therefore, it also allows a precise estimation of the orientation of the object.
2. We use MobileNetV3 to replace the original MobileNetV1 backbone in the skeleton extraction network, and we customize it to detect the skeleton of industrial objects with complex geometry from both front and back views, even if the objects have different shapes on either side.
3. The grasping pose selection is carried out using a dynamic approach. This means that an error in the skeleton extraction is autonomously detected and the robot actively modifies its position to better perform the grasping.

We have conducted several experiments with real-world automotive parts to validate our approach, which can detect the object's keypoints from upside-down views, without any constraint.

The remainder of the paper is organized as follows. Section 2 contains a brief description of existing related work. Our method is presented in Section 3. Experiments demonstrating the effectiveness of the proposed approach are shown in Section 4. Finally, conclusions and future directions are drawn in Section 5.

2. Related Work

In the last few years, thanks to the availability of powerful GPUs, deep learning methods have become suitable for dealing with grasping-point detection. They have proven capable of replacing traditional analytical approaches based on geometrical properties, physics models, and force analytics. A Convolutional Neural Network (CNN) architecture named GraspNet, able to segment graspable regions on the surfaces of objects, has been presented in [4]. In [5], a CNN, in combination with the information provided by a depth camera, has been used to detect the presence of the object and the best grasping pose. Several approaches were proposed to improve the accuracy of deep CNN, see, e.g., [6,7], but they usually require long computation time (i.e., of the order of seconds).

More efficient approaches, requiring only depth images, have been proposed in [8,9]. More in detail, in [8], a Deep Convolutional Neural Network has been trained in a simulated environment to learn grasping-relevant features and return a single-grasp solution for each object. In [9], the so-called generative grasping convolutional neural network (GG-CNN) has been proposed. It allows direct evaluation of the grasp quality and pose of grasps for every pixel in an input depth image, and it is fast enough to perform grasping in dynamic environments. The GG-CNN performance has been improved by introducing the GG-CNN2 [10], which is a CNN based on the semantic segmentation architecture of [11]. A common characteristic of deep-learning-based methods for grasping-point detection is the need to calculate the grasping quality value for each pixel in the image at hand, which is extremely time-consuming.

When the object knowledge and the grasp pose candidates are not available, it is possible to approximate the object using shape primitives, e.g., using multiview measurements [12] or identifying features in sensory data [13]. The method proposed in [14] consists of selecting grasp pose candidates after locating areas where a successful grasp had already been experienced. In [1], grasping partially known objects in unstructured environments is proposed based on an extension to the industrial context of the well-known technique of Background Subtraction [15]. Thanks to the spreading of low-cost depth sensors, many 3D registration algorithms have been exploited to handle the object grasping problem. For example, in [2], a model of the object to be grasped is generated using a set of point clouds acquired from different positions, and the nominal grasping pose is fixed. Subsequently, this model is compared with the runtime object view to compute the current grasping pose.

In this work, we propose a skeleton-based approach for detecting the grasping poses, which is inherently less computationally demanding due to the compact representation of the object via the skeleton.

A qualitative and quantitative comparison between our approach and the most relevant papers described in this section is shown in Table 1. This comparison takes into account not only the results but also the limitations of each work.

Table 1. Comparison table between different object grasping approaches.

Methods	Applications	Quantitative Results	Limitations
CNN architecture with a DDF module [4]	Real-time robotic grasping	90% of accuracy on Cornell grasp dataset	Error in predict orientation for some objects
Structure based on faster R-CNN and DACAB [5]	Object grasping with mobile manipulator	86.3% of success rate	Inefficient search method
GQ_CNN to classify robust grasping [6]	Grasping household objects	99% of precision	Long computational time
DCNN based on depth images to predict grasp pose [8]	Grasping of unknown objects	92% (70%) of precision with cylindrical-shaped (box-shaped) objects	Generation of a single-grasp solution for a single object
NN for learning prototypical parts [14]	Grasping of similar objects	N.A.	Grasping of complex-shaped objects with never-before-seen features.
Topological skeleton extraction (this work)	Grasping of complex-shaped automotive parts	87.04% of success rate	Needs a good camera calibration

3. Proposed Method

Figure 1 shows the overall functional architecture of our approach. It is made of four main modules, namely Visual Data Acquisition, Topological Skeleton Extraction, Grasping Pose Selection, and Robot Grasping. Each module is detailed below.

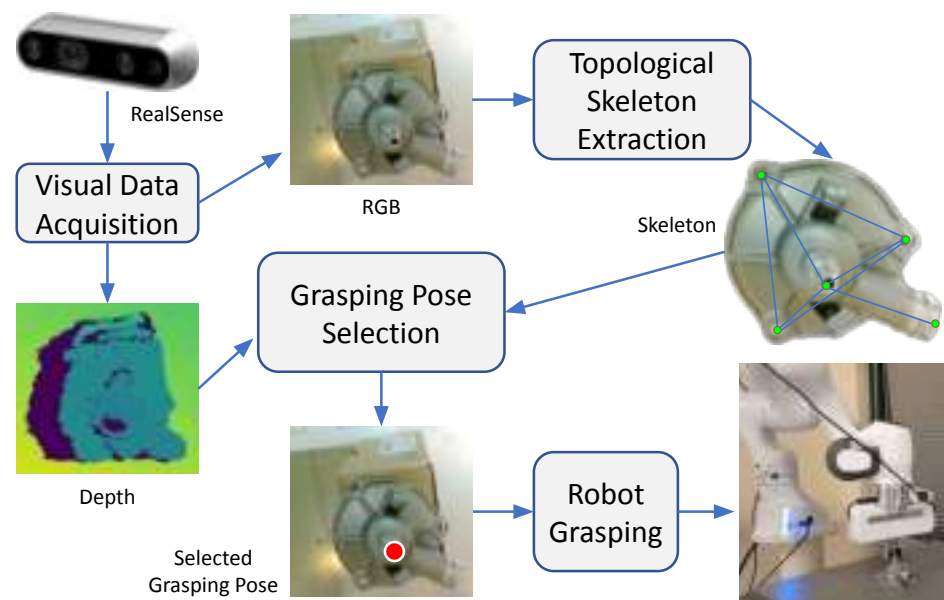


Figure 1. Functional architecture of the proposed approach.

3.1. Visual Data Acquisition

Visual data (both RGB and depth) are acquired using an Intel Realsense D435 RGBD camera, mounted on the end-effector via a 3D printed support in the so-called *eye-in-hand* configuration. Figure 2 shows the reference frames attached to the robot end-effector, $\mathcal{F}_e$, and the camera, $\mathcal{F}_c$.

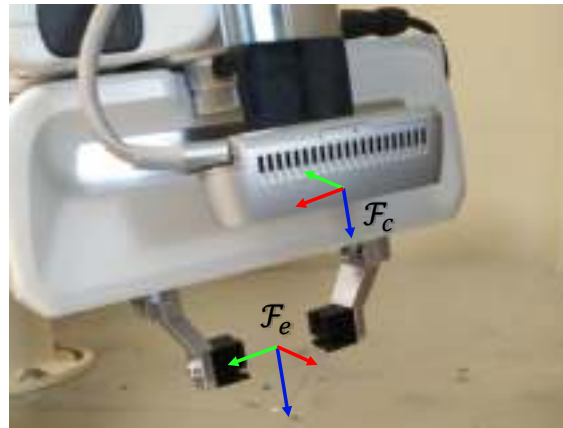


Figure 2. End-effector and camera reference frames.

The Intel Realsense D435 camera has a minimum depth distance beyond which it is not able to provide a depth measure approximately equal to 28 cm and the camera data acquisition requires the *realsense-ros* library since communication between the modules takes place through the Robot Operating System (ROS). The camera has been previously calibrated using 30 images of a 2D chessboard flat pattern. The calibration process includes both intrinsic and extrinsic calibrations. The first is aimed at determining the camera parameters that describe how the camera transforms the 3D coordinates of the scene into the 2D coordinates of the image, like the focal length, the principal point, and the optical distortions, while the second one provides the parameters which describe the rigid transformation that maps the 3D coordinates of the real world to the 3D coordinates of the camera's reference system. The calibration procedure implementation proposed by the VISP library [16], based on [17,18], has been adopted using a chessboard composed of 9×6 squares with dimensions of 0.02645 m.

It is worth remembering that a good calibration procedure is crucial for the success of the grasping procedure since it ensures an accurate perception of the environment, enabling precise identification and positioning of the points in three-dimensional space for successful manipulation.

3.2. Topological Skeleton Extraction

The proposed method has been developed for objects that:

- are rigid, as it is not applicable to deformable objects;
- are not perfectly symmetrical since although it is possible to define a non-symmetric topological skeleton, the detector may become confused during the extraction process due to symmetrical features.

In this work, we focus on real automotive parts, including two crankcase oil separator covers made of cast iron and plastic and an air pipe. The selected objects have an increasing level of difficulty. The first object, the cast iron crankcase oil separator cover, exhibits a high degree of symmetry with multiple grasping points and can be grasped by a cylindrical part, therefore reducing the impact of the robot orientation errors around the axis of the pin. The second object, the plastic crankcase oil separator cover, also exhibits a high degree of symmetry with various grasping points but must be grasped with a specific orientation. Finally, the air pipe has a complex shape, lacks symmetry, and has only two available grasping points, representing the most challenging task for the robot.

We decided to model their skeletons considering a few keypoints, some of which correspond to the potential grasping points for lifting that object with the robot manipulator (see the upper right part of Figure 1). To detect the Topological Skeleton (TS) of the objects to be grasped, we consider Lightweight OpenPose [19], whose architecture consists of three main components: a feature extractor, a TS estimator, and a Part Affinity Fields (PAF) network.

In comparison to the original OpenPose [20], we chose Lightweight OpenPose because the high computational demand of the former method makes it less applicable in real-time applications on devices with little processing power. OpenPose employs a two-branch, multi-stage CNN architecture. The first branch predicts part confidence maps (PCM) for body parts, and the second branch predicts part affinity fields (PAF) to model the connections between body parts. The architecture involves several stages of convolutions to refine these predictions iteratively, resulting in high accuracy at the cost of increased computational load. Light OpenPose, on the other hand, modifies the original architecture to reduce complexity and improve efficiency. The approach reduces the number of convolutional layers and stages and uses depthwise separable convolutions in place of standard convolutions to reduce the number of parameters and operations. Moreover, the backbone network uses MobileNet or ShuffleNet in place of the heavier VGG19 or ResNet used in the original OpenPose and optimizes the computation of part affinity fields to strike a balance between accuracy and efficiency.

Feature extraction. The original Lightweight OpenPose uses a MobileNetV1 network that is optimized for reaching real-time feature extraction. MobileNet is a family of neural network architectures designed for efficient deployment on mobile and embedded devices with limited computational resources. The key feature of MobileNet is its use of depthwise separable convolutions, which can significantly reduce the number of parameters and computations required while maintaining high accuracy.

While MobileNetV1 is a highly effective neural network, it does have some limitations and drawbacks that should be considered. For instance, it has limited accuracy because it is designed to balance model size and accuracy. It may not achieve the same level of accuracy as larger and more complex neural networks, especially on challenging objects where the keypoints (joints) are not evident. The depthwise separable convolution operation used in MobileNetV1 can be less expressive than traditional convolutional operations and may not be able to capture all the important features of an image.

For the above reasons, in this work, we propose to replace MobileNetV1 with MobileNetV3 [21] for the feature extraction step. MobileNetV3 has been designed to address the limitations of MobileNetV1 while maintaining efficiency. The architecture of the MobileNetV3 network used in this work is shown in Figure 3.

MobileNet V3 has two main variants: (1) *MobileNet V3-Large* designed for higher accuracy applications, with more layers and channels, and (2) *MobileNet V3-Small* optimized for resource-constrained environments, trading off some accuracy for reduced computational demand. Our choice fell on the latter one. MobileNet V3 introduces several new components, such as Inverted Residual Blocks to maintain a high degree of efficiency, Squeeze-and-Excitation (SE) Modules to improve the representational power of the model by recalibrating channel-wise feature responses and the H-Swish Activation Function.

The hard-swish function is a non-linear activation function that is designed to be more efficient than traditional activation functions such as ReLU. The hard-swish function is defined as

$$\text{h-swish}(x) = x \frac{\text{ReLU6}(x + 3)}{6}, \quad (1)$$

where $\text{ReLU6}(x) = \min(\max(x, 0), 6)$ is a clipped ReLU function that outputs values between 0 and 6, still providing a non-linear behavior while increasing the computational speed with respect to the standard ReLU function.

Another important feature of MobileNetV3 is the use of a squeeze-and-excitation (SE) module. The SE module is a simple and efficient way to improve the representational power

of the network (i.e., the ability to learn and represent complex patterns and features in the input data). It works by learning channel-wise scaling factors that are used to selectively enhance informative features in the network. The SE module is added to each bottleneck block in the MobileNetV3 architecture, contributing to increasing the accuracy with respect to MobileNetV1.

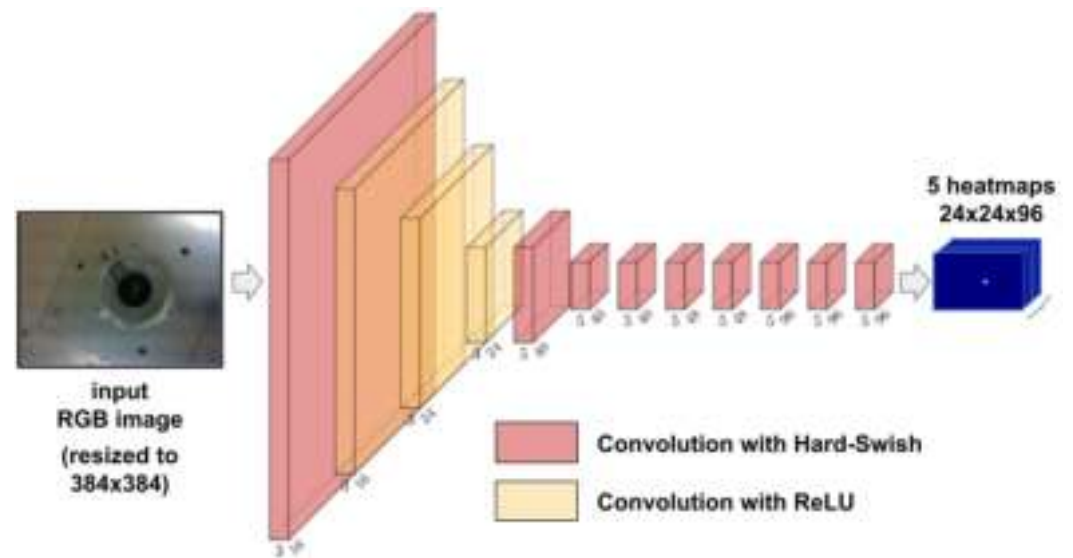


Figure 3. MobileNetV3 architecture.

MobileNetV3 also introduces a new technique called the mobile inverted bottleneck convolution (MBConv), which is a modified form of the depthwise separable convolution used in MobileNetV1. The MBConv block consists of three types of convolutions: a 1×1 convolution to expand the number of channels, a depthwise convolution to perform spatial filtering, and a 1×1 convolution to reduce the number of channels back to the original size. The MBConv block also includes a shortcut connection that allows the gradient to flow directly from the input to the output. MBConv block helps in increasing the expressiveness of the model with respect to MobileNetV1.

Finally, MobileNetV3 includes a middle-flow block that is used to maintain a high level of accuracy while minimizing the number of computations required. It also uses a dynamic convolution operation that adapts to the input data. The details of the parameters used for each block are described in Table 2. The input image is resized to 384×384 , and the output is a set of $24 \times 24 \times 96$ feature maps, one for each keypoint and one for the background.

Table 2. The MobileNetV3 network architecture used in this paper. HS = hard-swish, RE = ReLU, s = stride.

Input	Operator	Exp Size	#out	SE	NL	s
$384^2 \times 3$	conv2d, 3×3	-	16	-	HS	2
$192^2 \times 16$	bneck, 3×3	16	16	x	RE	2
$96^2 \times 16$	bneck, 3×3	72	24	-	RE	2
$48^2 \times 24$	bneck, 3×3	88	24	-	RE	1
$48^2 \times 24$	bneck, 5×5	96	40	x	HS	2
$24^2 \times 40$	bneck, 5×5	240	40	x	HS	1
$24^2 \times 40$	bneck, 5×5	240	40	x	HS	1
$24^2 \times 40$	bneck, 5×5	120	48	x	HS	1
$24^2 \times 48$	bneck, 5×5	144	48	x	HS	1
$24^2 \times 48$	bneck, 5×5	288	96	x	HS	1
$24^2 \times 96$	bneck, 5×5	576	96	x	HS	1
$24^2 \times 96$	bneck, 5×5	576	96	x	HS	1

TS estimation. The feature maps from MobileNetV3 are the input to generate a set of candidate key points for each object part in the image. In fact, the feature maps capture the spatial information in the input image and provide a rich representation of the image that can be used to detect keypoints. MobileNetV3 adds a custom head to predict keypoint locations, which consists of a series of convolutional layers that generate heatmaps, refining the features extracted by the backbone and generating heatmaps for each keypoint. Figure 4 shows an example of the TS estimator output for the cast iron crankcase oil separator cover, which consists of five heatmaps, one for each considered keypoint. Each heatmap has the same spatial resolution as the feature maps and is normalized to have values between 0 and 1. Each pixel in the heatmap indicates the likelihood that the corresponding body part is present at that location in the image.

PAF network. It takes the feature maps generated by the feature extractor as input and outputs a set of PAF feature maps, one for each pair of the detected keypoints. The PAF feature maps encode the direction and strength of the connections between keypoints using a two-channel representation, where each channel encodes a different aspect of the connection. Specifically, one channel encodes the unit vector that represents the direction of the connection, while the other channel encodes the confidence score that represents the strength of the connection.

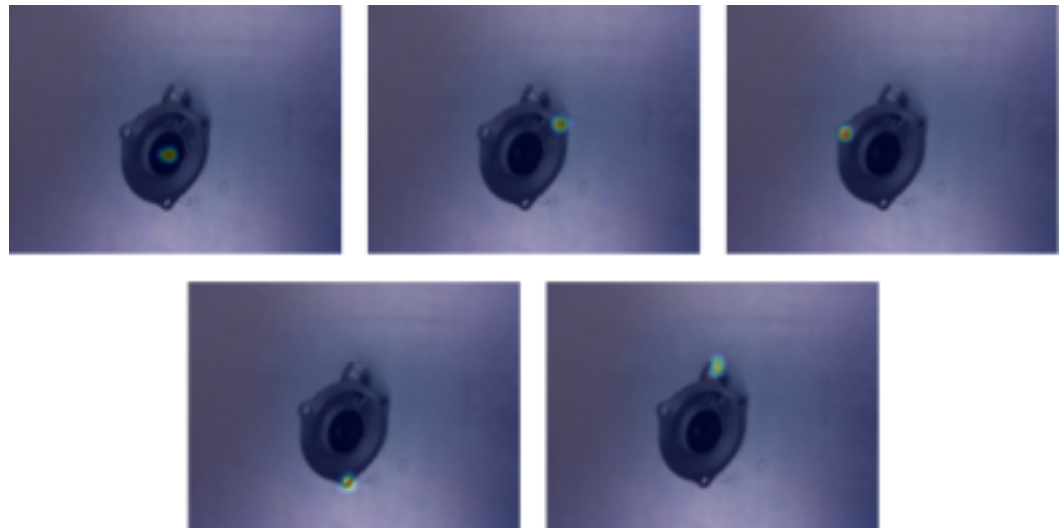


Figure 4. Heatmap examples for cast iron crankcase oil separator cover. There are five heatmaps corresponding to the considered keypoints.

Final TS computation. Once the PAF and heatmaps are generated, they are used together to group the individual keypoints into the final TS. The final TS is obtained by first identifying the candidate connections using the PAFs and then scoring the connections based on the likelihood that they form a valid connection. The connections are then used to construct the final TS by connecting the individual keypoints into a complete object TS.

Once the keypoints are calculated, we use the depth information for building the final 3D TS given the set of keypoints from Lightweight OpenPose. Figure 5 shows some examples of final TSs for the three objects considered, where it is possible to note the robustness of the proposed TS extraction approach with respect to different views of the object, to photochromic changes and partial occlusions. The approach is also working with multiple instances of the object.

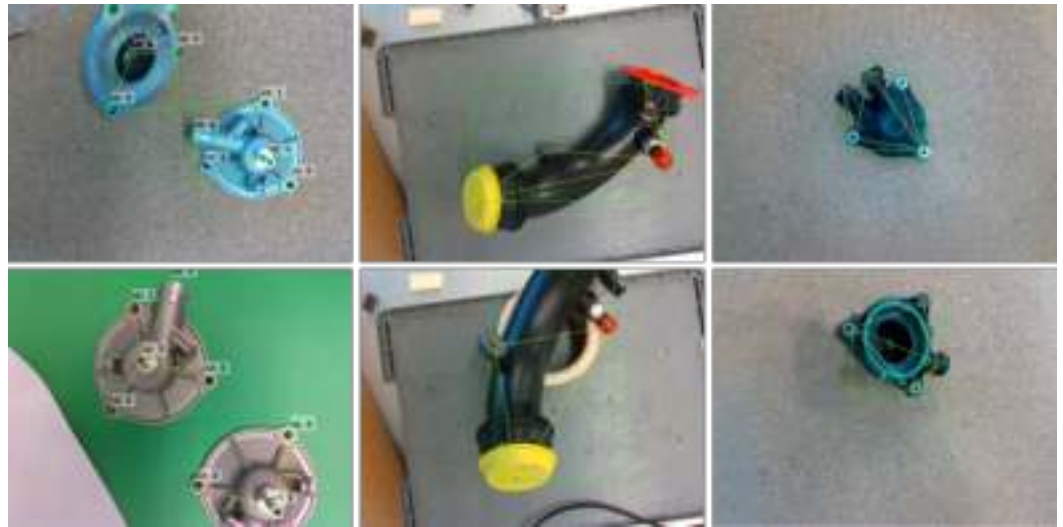


Figure 5. TS extraction examples on different objects: the cast iron crankcase oil separator cover on the left, the air pipe in the middle, and the plastic crankcase oil separator cover on the right. Our approach is robust to different views of the same object, to photochromic changes, and partial occlusions.

3.3. Grasping Pose Selection

After the selection of the N_k keypoints for the TS extraction, these keypoints are also identified within the CAD model of the object through 3D modeling software. This results in the generation of a nominal three-dimensional representation of the TS, TS_N , in the CAD coordinate system, $\mathcal{F}_f$. Moreover, the poses in $\mathcal{F}_f$ of all possible N_g grasping reference frames (see Figure 6), expressed via the (4×4) homogeneous transformation matrix [22], $T_{g_j}^f, j = 1, \dots, N_g$, can be localized on the model. For the sake of clarity, let us assume that each grasping point coincides with a keypoint.



Figure 6. The grasping frames for the considered objects: cast iron crankcase oil separator cover (**top row**), plastic crankcase oil separator cover (**middle row**), and air pipe (**bottom row**).

Then, given all possible combinations of three keypoints

$$\mathcal{S}_t = \left\{ t_i, i = 1, \dots, N_t = \binom{N_k}{3} : t_i = (P_j, P_l, P_m), j, l, m \in \{1, \dots, N_k\}, j \neq l \neq m \right\},$$

for each triple t_i , a plane is identified via a coordinate frame attached to it, whose pose is denoted by the homogeneous transformation matrix $T_{t_i}^f$. For each grasping reference frame, the relative pose with respect to the i th plane can be determined as

$$T_{g_j}^{t_i} = \left(T_{t_i}^f\right)^{-1} T_{g_j}^f. \quad (2)$$

Thus, for each grasping point, a list of N_t transformation matrices, $T_{g_j}^{t_i}$, representing the grasping frame poses in the plane frame, can be computed. This set of operations, summarized in Algorithm 1, is performed only once.

Algorithm 1: Pre-processing algorithm

Input : $TS_N, N_g, N_t, T_{g_j}^f (j = 1, \dots, N_g)$
Output: $T_{g_j}^{t_i} (i = 1, \dots, N_t; j = 1, \dots, N_g)$
1 **for** each triple t_i of keypoints in TS_N **do**
2 Compute $T_{t_i}^f$
3 **end for**
4 **for** $i = 1, \dots, N_t$ **do**
5 **for** $j = 1, \dots, N_g$ **do**
6 $T_{g_j}^{t_i} = \left(T_{t_i}^f\right)^{-1} T_{g_j}^f$
7 **end for**
8 **end for**
9 **return** $T_{g_j}^{t_i}$

At runtime, the following steps are executed:

1. A YOLO detector [23] is adopted to distinguish between the objects. YOLO has been chosen since it is faster than classifier-based systems but with similar accuracy and makes predictions with a single network evaluation by considering object detection as a single regression problem, and this leads to high accuracy performance. Moreover, YOLO can detect and classify multiple objects simultaneously within an image.
2. The current 3D TS, TS_C , is extracted.
3. The grasping point closest to the camera, p_{cc}^c , is selected as the best one.
- 4a. If at least 3 keypoints are visible, a set of three keypoints, t_k , is used to compute the corresponding plane in the camera frame, $T_{t_k}^c$, and to select the homogeneous transformation matrix, $T_{g_{cc}}^{t_k}$, that identifies the grasping pose in the plane frame. Then, the procedure continues with the step 5.
- 4b. If only 2 or fewer keypoints are visible, the robot starts moving in a circle around the center of the object bounding box to acquire a new image from a different point of view. Then, the procedure comes back to the step 1.
5. The grasping pose in camera frame is computed as

$$T_{g_{cc}}^c = T_{t_k}^c T_{g_{cc}}^{t_k}.$$

This procedure is summarized in Algorithm 2.

Let us define the homogeneous transformation matrix T_c^e , i.e., the constant homogeneous matrix performing the transformation between the camera frame and the end-effector frame, obtained via the calibration method described in Section 3.1.

Algorithm 2: Runtime algorithm

Input : $obj T_{g_j}^{t_i}$ for each object
Output: $T_{g_{cc}}^c$

- 1 Get image from camera
- 2 Detect current object from image
- 3 $TS_C \leftarrow$ extract TS
- 4 $\mathcal{T} = T_{g_j}^{t_i} \Big|_{i=1, \dots, N_t; j=1, \dots, N_g} \leftarrow$ get the list of transformations relative to the current object
- 5 $p_{cc}^c \leftarrow$ extract the grasping point closest to the camera
- 6 **if** $visible\_keypoints \geq 3$ **then**
 - 7 $t_k \leftarrow$ extract a triple of visible keypoints in TS_C
 - 8 Compute $T_{t_k}^c$
 - 9 Extract $T_{g_{cc}}^{t_k}$ from $\mathcal{T}$
- 10 **else**
 - 11 move the robot to a different point of view
 - 12 GO TO 1
- 13 **end if**
- 14 $T_{g_{cc}}^c \leftarrow T_{t_k}^c T_{g_{cc}}^{t_k}$
- 15 **return** $T_{g_{cc}}^c$

To capture the grasping pose in the inertial frame, $T_{g_{cc}}^c$ is transformed as follows

$$T_{g_{cc}} = T_e T_c^e T_{g_{cc}}^c, \quad (3)$$

where T_e is the homogeneous matrix representing the pose of the end-effector in the inertial frame.

Remark 1. It is worth noting that if the grasping point is not coincident with a keypoint, the above procedure is still applicable, but a further constant transformation needs to be applied to link the grasping point to one of the keypoints belonging to the plane.

3.4. Robot Grasping

To perform the grasp, the end-effector must be commanded to align its reference frame to the grasping reference frame. The trajectory is planned by assigning a sequence of three points: the first one is the view pose of the robot, the intermediate one is the *approach* point, i.e., a point positioned along the z axis of the grasping reference frame at a distance of 10 cm to the origin, and the last one is the estimated grasping position, $\hat{p}_g$. More in detail, the end-effector desired position, $p_{e,d}(t)$, is defined as

$$p_{e,d}(t) = \begin{cases} p_0 + \frac{s_1(t)}{\|p_a - p_0\|} (p_a - p_0) & \text{for } 0 \leq t \leq t_a \\ p_a + \frac{s_2(t)}{\|\hat{p}_g - p_a\|} (\hat{p}_g - p_a) & \text{for } t_a < t \leq t_f \end{cases}. \quad (4)$$

where p_0 is the view position and p_a is the approach point position, $s_1(t)$ ($s_2(t)$) is the *arc length* from p_0 to p_a (from p_a to $\hat{p}_g$). To ensure continuous acceleration and velocities at the path points, both for $s_1(t)$ and $s_2(t)$, the time-law can be designed as a quintic polynomial. Regarding the time instants, t_f is the duration of the motion, and t_a is the intermediate time instant at the approach point that is chosen to have a fast motion until the approach point and a slow motion in the object's proximity.

Regarding the end-effector orientation, it is planned to reach the same orientation of the estimated grasping pose, $\hat{R}_g$, at the approach point and to keep such orientation constant during the last part of the path.

The planned trajectory in terms of position and orientation is the input of the closed-loop inverse kinematics algorithm [22] aimed at computing the reference values of the joint positions and velocities. Let denote with $p_e(t)$ and $R_e(t)$ the end-effector position and orientation, respectively, and with $R_{e,d}(t)$ the end-effector desired orientation. The robot joint velocity references, $\dot{q}_r(t)$, are computed as

$$\dot{q}_r(t) = J^+(q(t))(\dot{v}_{e,d}(t) + Ke(t)), \quad (5)$$

where $J^+(q(t))$ denotes the right pseudo-inverse of the robot Jacobian matrix, $K \in \mathbb{R}^{6 \times 6}$ is a positive definite matrix gain, $v_{e,d} = [\dot{p}_{e,d}^T \ \omega_{e,d}^T]^T$ is the desired end-effector linear and angular velocity, and e is the tracking error defined as

$$e = \begin{bmatrix} p_{e,d} - p_e \\ \eta_e \epsilon_{e,d} - \eta_{e,d} \epsilon_e - S(\epsilon_{e,d}) \epsilon_e \end{bmatrix}, \quad (6)$$

where $Q_e = \{\eta_e, \epsilon_e\}$ and $Q_{e,d} = \{\eta_{e,d}, \epsilon_{e,d}\}$ are the unit quaternion extracted from R_e and $R_{e,d}$, respectively, and $S(\cdot)$ is the skew-symmetric matrix operator performing the cross product [22].

A flowchart representation highlighting the whole process is given in Figure 7.

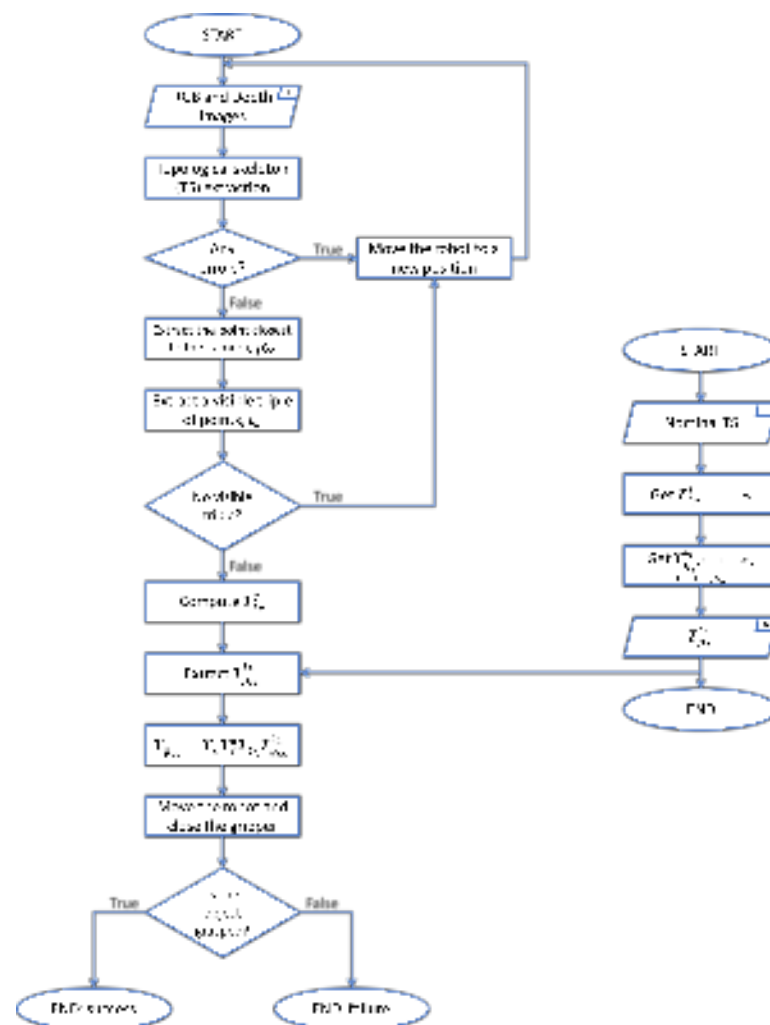


Figure 7. Flowchart representation of the whole process.

4. Experimental Results

The experimental setup consists of an Intel RealSense D435 camera mounted on a Franka Emika Panda robot manipulator, characterized by 7 revolute joints. The robot can be controlled by means of the Franka Control Interface (FCI) and the `libfranka` C++ open-source library, which directly controls the robot with an external workstation through an ethernet connection. In this work, the `franka_ros` meta-package, which integrates `libfranka` into ROS, has been used. The workstation runs Ubuntu 18.04 LTS and a real-time kernel on an Intel Xeon 3.7 GHz CPU with 32 GB RAM. We have conducted experiments with the three considered objects shown in Figure 6, and the quantitative results are reported below.

4.1. TS Extraction Results

Using Coco Annotator [24], 5992 images have been annotated. The labeled data have been split into Training, Validation, and Test sets composed of 4618, 229, and 1145 images, respectively. Table 3 shows the number of images in the Training, Validation, and Test sets for each considered object.

Table 3. Number of sample images used in Training, Validation, and Test sets for the considered objects.

Object	Training	Validation	Test
Cast iron crankcase oil separator cover	1440	60	300
Air pipe	1406	46	228
Plastic crankcase oil separator cover	1772	123	617

The metric we used for evaluating the TS detection is the Object Keypoint Similarity (OKS) [25], defined as follows:

$$OKS = \frac{\sum_{i \in [0, N-1]} \exp\left(\frac{-d_i^2}{2s^2k_i^2}\right) \delta(v_i > 0)}{\sum_{i \in [0, N-1]} \delta(v_i > 0)} \quad (7)$$

where:

- s is the object scale;
- d_i is the distance of the predicted keypoint i from the ground truth;
- k_i is a per-keypoint constant that controls the falloff;
- v_i is the visibility flag.

OKS is calculated for each sample representing an object. The visibility flag takes into account if a point is visible or not: if the keypoint is labeled, $\delta(v_i > 0)$ is 1, else it is 0 without considering occluded keypoints.

In our scenario, we used OKS to compute the True Positive (TP), False Negative (FN), False Positive (FP), and True Negative (TN) detections. If a detection has $OKS > threshold$, it is considered to be a TP; otherwise, as an FP. In particular, we considered two thresholds, namely 0.5 and 0.75, and calculated the following metrics: Precision, Recall, F1-score, and Average Precision (AP). Table 4 shows the results of our algorithm for a test set of 1145 images.

Table 4. Results of the TS detector at different thresholds for a test set of 1145 images.

Threshold	Precision	Recall	F1-Score	AP
0.5	0.92	0.90	0.91	0.82
0.75	0.86	0.89	0.87	0.72

To compute the runtime performance of our TS extractor, we tested it on a subset of 60 images using an NVIDIA RTX A5500, obtaining an average execution time of 0.012 s and

a standard deviation of 0.0018 s. On a subset of 40 images, using an NVIDIA QUADRO T2000, the average execution time is 0.019 s, and the standard deviation is 0.0025 s.

4.2. Object Detector Results

For training the object detector, we annotated 750 images of size 640×480 using the LabelImg annotation tool [26]. We split the dataset into Train, Validation, and Test sets composed of 450, 150, and 150 images, respectively. After the training stage, the mean average precision on the test set is 97.32%, and the success rate is 96.70%. The inference on the images has been executed on an NVIDIA QUADRO T2000. On a subset of 40 images, the average execution time is 0.323 s, while the standard deviation is 0.0615 s.

4.3. Robot Grasping Results

Let us define the estimation grasping position and orientation errors as

$$e_p^e = p_g^e - \hat{p}_g^e, \quad (8)$$

$$e_\phi^e = \phi_g^e - \hat{\phi}_g^e, \quad (9)$$

where p_g^e is the actual grasping position while $\hat{p}_g^e$ is the estimate provided by the visual algorithm. Regarding the orientation, ϕ_g^e ($\hat{\phi}_g^e$) is the Euler angles extracted from the actual (estimated) grasping pose. The adoption of Euler angles in lieu of quaternions as in (6) provides a clearer physical interpretation of the orientation errors. The superscript e denotes that the variables are expressed in the end-effector frame (see Figure 2).

To have statistically significant results, 54 grasping tests (20 for the cast iron crankcase oil separator cover, 19 for the air pipe, and 15 for the plastic crankcase oil separator cover) have been conducted by placing the objects in different configurations, different light conditions, and with different backgrounds in a way to let the robot explore all the possible grasping poses. A grasping test is considered successful if the gripper holds the object with a stable grasping for 10 s. A set of snapshots of the grasping procedure is shown in Figure 8, where the top row refers to a successful test and the bottom row refers to a failure.

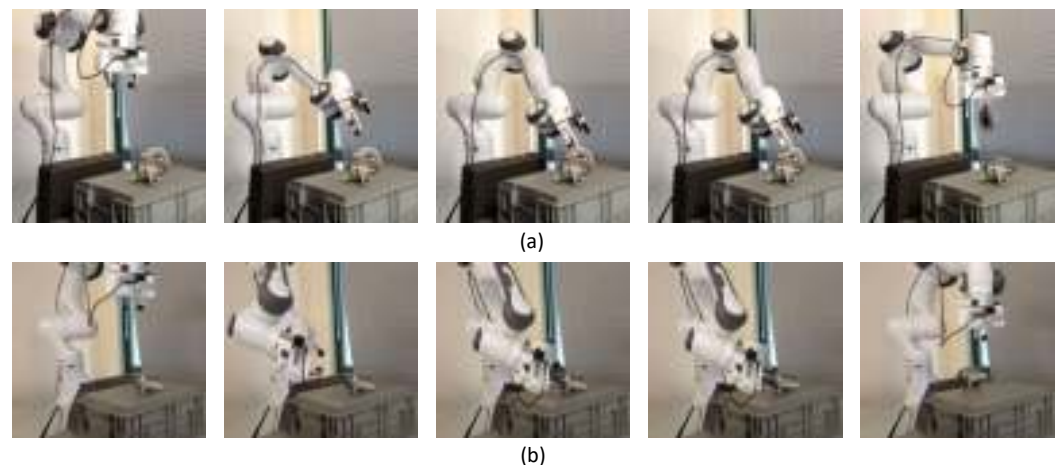


Figure 8. Snapshots of two grasping cases. (a) Successful grasp. (b) Failure.

Only 7 experiments (2 for the cast iron crankcase oil separator cover, 3 for the air pipe, and 2 for the plastic crankcase oil separator cover) experienced a failure. Thus, a success rate of 87.04% has been obtained. Tables 5–7 show the mean position and orientation errors and the corresponding standard deviation for the successful tests.

Table 5. Mean errors for the cast iron crankcase oil separator cover.

Successful Tests	$e_{p_x}^e$ [mm]	$e_{p_y}^e$ [mm]	$e_{\phi_x}^e$ [deg]	$e_{\phi_y}^e$ [deg]
1	0.734	9.471	3.128	3.515
2	6.867	0.717	3.284	3.091
3	1.081	5.857	6.092	12.433
4	13.032	10.752	3.132	6.256
5	1.796	0.775	6.801	0.77
6	3.832	0.759	3.913	3.546
7	3.629	1.531	2.981	3.416
8	1.324	6.747	0.674	0.92
9	3.073	7.437	0.62	35.308
10	1.293	1.021	0.952	1.605
11	3.916	1.631	0.391	6.217
12	4.62	2.697	3.044	4.508
13	2.368	4.09	9.12	3.953
14	0.463	6.314	2.456	10.075
15	1.816	2.214	2.217	5.806
16	2.958	4.751	1.842	7.763
17	2.86	8.736	1.354	4.138
18	3.018	0.747	13.855	2.16
Mean	3.26	4.236	3.659	6.415
Standard deviation	2.820	3.278	3.329	7.607

Table 6. Mean errors for the air pipe.

Successful Tests	$e_{p_x}^e$ [mm]	$e_{p_y}^e$ [mm]	$e_{\phi_x}^e$ [deg]	$e_{\phi_y}^e$ [deg]
1	0.043	0.989	2.439	8.109
2	9.586	17.223	4.597	0.555
3	1.098	2.612	16.723	13.204
4	3.777	6.188	19.545	10.702
5	7.213	0.227	3.663	1.042
6	3.882	1.516	1.684	0.315
7	2.897	2.176	11.183	1.733
8	0.144	1.932	10.255	7.891
9	2.41	1.412	12.057	7.392
10	5.042	0.797	12.674	20.755
11	2.989	2.809	11.516	26.216
12	14.5	4.903	0.988	2.294
13	11.149	1.207	5.192	0.53
14	3.736	5.841	9.211	25.563
15	0.818	8.494	12.022	20.307
16	0.636	14.538	15.122	7.086
Mean	4.37	4.554	9.304	9.606
Standard deviation	4.085	4.843	5.447	8.801

For all the objects, the position errors along the z -axis of the end-effector frame have not been reported since they are negligible due to the object geometry. For the same reason, the orientation errors around the z -axis of the end-effector frame can be negligible for the cast iron crankcase oil separator cover and the air pipe.

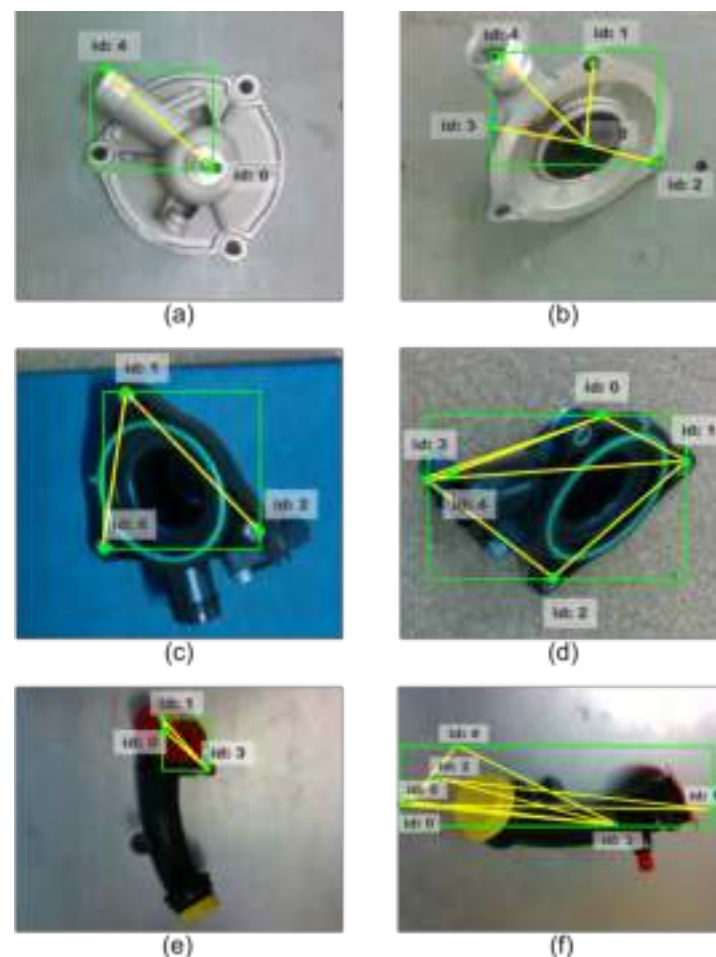
In some tests, large errors have been experienced, mostly along the y -axis of the end-effector frame, but the object has been successfully grasped since the gripper is characterized by parallel fingers, and errors along the closing direction are more tolerated.

The system failures can be divided into two main categories:

1. Errors related to missing (see Figure 9a,e) or inaccurate (see Figure 9b,d,f) keypoint detection or prediction, and wrong depth estimation.
2. Pose estimation errors that can cause the slipping of the object.

Table 7. Mean errors for the plastic crankcase oil separator cover.

Successful Tests	$e_{p_x}^e$ [mm]	$e_{p_y}^e$ [mm]	$e_{\phi_x}^e$ [deg]	$e_{\phi_y}^e$ [deg]	$e_{\phi_z}^e$ [deg]
1	13.453	1.036	7.726	4.594	10.505
2	1.795	8.396	11.419	16.797	1.977
3	1.387	5.306	2.214	3.451	11.761
4	7.493	5.909	3.41	5.945	7.285
5	0.799	6.753	3.439	12.744	2.62
6	2.997	1.521	2.03	4.885	9.134
7	5.897	9.452	10.56	1.247	4.62
8	0.973	1.319	7.221	0.007	4.146
9	2.246	5.577	2.104	3.025	2.231
10	1.268	13.631	3.325	4.165	26.108
11	4.547	13.786	2.265	1.418	7.61
12	0.575	1.472	8.638	2.531	10.629
13	3.606	2.318	1.721	6.289	23.799
Mean	3.618	5.883	5.082	5.161	9.417
Standard deviation	3.487	4.281	3.381	4.527	7.367

**Figure 9.** Examples of missing (a,e) and inaccurate (b,d,f) keypoint detection in TS. Example of missing keypoint detection that can lead to a successful object grasping (c).

In the case of a missing keypoint detection, e.g., due to the relative object-camera position, the failure can be managed by moving the camera's point of view and acquiring a new prediction (see Section 3.3). In the other cases, the grasping procedure is completed with a failure. Since the robot can detect the grasping failure, the whole process is repeated.

It is worth noticing that, according to the procedure outlined in Algorithm 2, the grasping of an object is feasible even with only three visible keypoints (see Figure 9c) correctly detected, provided that one of these is a grasping point situated in a location that can be grasped with the available end-effector.

5. Conclusions

In this work, a robust method for complex-geometry parts grasping in an industrial scenario has been proposed. In such an environment, grasping challenges are due to the presence of uncertainties in the position of the object to grasp and to the perception of noise due to its material. In particular, we focused on real-world automotive parts with complex geometries and reflective surfaces that provoke noise in the depth map. The proposed solution relies on a TS extraction network that can create a graph-based representation of the object in real time. A reasoning step is used to decide if the current view of the object is good enough for the actual grasping or if the manipulator needs to move to better grasp the object. Quantitative experiments have been conducted with a 7 DoF robot and three different complex-shaped automotive parts, demonstrating that the proposed approach is fast and robust. The high accuracy and real-time capability of this proposed approach render it a suitable solution for industrial applications where fast and accurate performance is required.

Due to the complexity of the considered objects, a complete quantitative performance comparison with other approaches present in the literature can hardly be carried out. However, the test dataset is publicly available to make possible future comparisons.

In future directions, we intend to study the integration of the depth data inside the TS extraction process to directly obtain the 3D positions of the keypoints. Moreover, the object detection phase could also be integrated into the TS extraction procedure.

Author Contributions: Conceptualization, A.P., M.S., D.D.B. and F.P.; Methodology, A.P., M.S., D.D.B. and F.P.; Software, A.P., M.S. and D.D.B.; Investigation, A.P. and M.S.; Validation, A.P. and M.S.; Formal analysis, A.P., M.S. and D.D.B.; Writing—original draft, A.P. and M.S.; Writing—review and editing, D.D.B. and F.P.; Supervision, D.D.B. and F.P.; Project administration, D.D.B. and F.P.; Funding acquisition, F.P. All the authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Italian Ministry of University and Research under the grant PRIN 2022 PNRR MELODY (Multi-robot collaborativE manipuLation suppOrting Disassembly tasks) n. P2022XALNS.

Data Availability Statement: The source code of the TS detector approach is publicly available at <https://github.com/apennisi/CoGP-TS>. The ROS-based source code of our approach is publicly available at <https://github.com/sileom/graspingWithSkeleton.git>. Several videos of the experiments are available at <https://tinyurl.com/bdhyf493>. The test set images are available at <https://tinyurl.com/bdxs8n7z>. All the models used in the described strategy are publicly available and can be downloaded from <https://tinyurl.com/3a4nnc88> (All links accessed on 23 July 2024).

Acknowledgments: The authors would like to thank Alessandro Lorenzo, Simona D’Amato, and Antonio Giardiello for their help with the image annotation process.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
DoF	Degree of Freedom
FCI	Franka Control Interface
FCN	Fully Convolutional Network
FN	False Negative

FP	False Positive
GG-CNN	Generative Grasping Convolutional Neural Network
MBConv	Mobile inverted Bottleneck Convolution
OKS	Object Keypoint Similarity
PAF	Part Affinity Fields
ROS	Robot Operating System
TN	True Negative
TP	True Positive
TS	Topological Skeleton

References

1. Sileo, M.; Bloisi, D.D.; Pierri, F. Real-time Object Detection and Grasping Using Background Subtraction in an Industrial Scenario. In Proceedings of the 2021 IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI), Virtual, 6–9 September 2021; pp. 283–288.
2. Sileo, M.; Bloisi, D.D.; Pierri, F. Grasping of Solid Industrial Objects Using 3D Registration. *Machines* **2023**, *11*, 396. [\[CrossRef\]](#)
3. Costanzo, M.; De Maria, G.; Lettera, G.; Natale, C. Can robots refill a supermarket shelf?: Motion planning and grasp control. *IEEE Robot. Autom. Mag.* **2021**, *28*, 61–73. [\[CrossRef\]](#)
4. Asif, U.; Tang, J.; Harrer, S. GraspNet: An Efficient Convolutional Neural Network for Real-time Grasp Detection for Low-powered Devices. In Proceedings of the IJCAI, Stockholm, Sweden, 13–19 July 2018; pp. 4875–4882.
5. Zhang, H.; Tan, J.; Zhao, C.; Liang, Z.; Liu, L.; Zhong, H.; Fan, S. A fast detection and grasping method for mobile manipulator based on improved faster R-CNN. *Ind. Robot. Int. J. Robot. Res. Appl.* **2020**, *47*, 167–175. [\[CrossRef\]](#)
6. Mahler, J.; Liang, J.; Niyaz, S.; Laskey, M.; Doan, R.; Liu, X.; Ojea, J.A.; Goldberg, K. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In Proceedings of the Robotics: Science and Systems (RSS), Cambridge, MA, USA, 12–16 July 2017.
7. Pinto, L.; Gupta, A. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In Proceedings of the 2016 IEEE international conference on robotics and automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 3406–3413.
8. Schmidt, P.; Vahrenkamp, N.; Wächter, M.; Asfour, T. Grasping of unknown objects using deep convolutional neural networks based on depth images. In Proceedings of the 2018 IEEE international conference on robotics and automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 6831–6838.
9. Morrison, D.; Corke, P.; Leitner, J. Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach. In Proceedings of the Robotics: Science and Systems (RSS), Pittsburgh, PA, USA, 26–30 June 2018.
10. Morrison, D.; Corke, P.; Leitner, J. Learning robust, real-time, reactive robotic grasping. *Int. J. Robot. Res.* **2020**, *39*, 027836491985906. [\[CrossRef\]](#)
11. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2015**, arXiv:1511.07122.
12. Dune, C.; Marchand, E.; Collin, C.; Leroux, C. Active rough shape estimation of unknown objects. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and System, Nice, France, 22–26 September 2008; pp. 3622–3627.
13. Kraft, D.; Pugeault, N.; Başeski, E.; POPOVIĆ, M.; Kragić, D.; Kalkan, S.; Wörgötter, F.; Krüger, N. Birth of the object: Detection of objectness and extraction of object shape through object–action complexes. *Int. J. Humanoid Robot.* **2008**, *5*, 247–265. [\[CrossRef\]](#)
14. Detry, R.; Ek, C.H.; Madry, M.; Piater, J.; Kragic, D. Generalizing grasps across partly similar objects. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, St Paul, MI, USA, 14–18 May 2012; pp. 3791–3797.
15. Bloisi, D.D.; Pennisi, A.; Iocchi, L. Background modeling in the maritime domain. *Mach. Vis. Appl.* **2014**, *25*, 1257–1269. [\[CrossRef\]](#)
16. Marchand, É.; Spindler, F.; Chaumette, F. ViSP for visual servoing: A generic software platform with a wide class of robot control skills. *IEEE Robot. Autom. Mag.* **2005**, *12*, 40–52. [\[CrossRef\]](#)
17. Kannala, J.; Brandt, S.S. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1335–1340. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Tsai, R.Y.; Lenz, R.K. A new technique for fully autonomous and efficient 3 D robotics hand/eye calibration. *IEEE Trans. Robot. Autom.* **1989**, *5*, 345–358. [\[CrossRef\]](#)
19. Osokin, D. Real-time 2D Multi-Person Pose Estimation on CPU: Lightweight OpenPose. *arXiv* **2018**, arXiv:1811.12004.
20. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.E.; Sheikh, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 172–186. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Howard, A.; Sandler, M.; Chu, G.; Chen, L.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. *arXiv* **2019**, arXiv:1905.02244.
22. Siciliano, B.; Sciacivico, L.; Villani, L.; Oriolo, G. *Robotics—Modelling, Planning and Control*; Springer: London, UK, 2009.
23. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
24. Brooks, J. COCO Annotator. 2019. Available online: <https://github.com/jsbroks/coco-annotator/> (accessed on 23 July 2024).

25. Ronchi, M.R.; Perona, P. Benchmarking and Error Diagnosis in Multi-Instance Pose Estimation. *arXiv* **2017**, arXiv:1707.05388.
26. Heartexlabs; Lin, T. LabelImg. 2015. Available online: <https://github.com/heartexlabs/labelImg> (accessed on 23 July 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Improved Active Disturbance Rejection Control for Permanent Magnet Synchronous Motor

Zhiwei Huang¹, Yuanhao Cheng^{1,2,*}, Si Chen^{1,2}, Xuhui Zhang³ , Jiawei Xiang^{1,2}  and Sun'an Wang⁴ 

¹ College of Mechanical and Electrical Engineering, Wenzhou University, Wenzhou 325060, China; 22461440036@stu.wzu.edu.cn (Z.H.); 20200316@wzu.edu.cn (S.C.); jwxiang@wzu.edu.cn (J.X.)

² Pingyang Institute and Intelligent Manufacturing, Wenzhou University, Wenzhou 325035, China

³ Shaanxi Key Laboratory of Mine Electromechanical Equipment Intelligent Monitoring, Xi'an University of Science and Technology, Xi'an 710064, China; zhangxh@xust.edu.cn

⁴ School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, China; sawang@xjtu.edu.cn

* Correspondence: yhcheng@wzu.edu.cn

Abstract: To improve the control performance of a permanent magnet synchronous motor (PMSM) under external disturbances, an improved active disturbance rejection control (IADRC) algorithm is proposed. Since the nonlinear function in the conventional ADRC algorithm is not smooth enough at the breakpoints, which directly affects the control performance, an innovative nonlinear function is proposed to effectively improve the convergence and stability. On this basis, the proposed IADRC is constructed, and comparative simulation results with ADRC and other IADRC show that faster response speed, higher accuracy and stronger robustness are obtained.

Keywords: improved ADRC; nonlinear function; PMSM; speed control



Citation: Huang, Z.; Cheng, Y.; Chen, S.; Zhang, X.; Xiang, J.; Wang, S. Improved Active Disturbance Rejection Control for Permanent Magnet Synchronous Motor.

Electronics **2024**, *13*, 3023. <https://doi.org/10.3390/electronics13153023>

Academic Editor: Fabio Corti

Received: 25 June 2024

Revised: 25 July 2024

Accepted: 30 July 2024

Published: 31 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Due to the advantages of high power density, simple structure, easy maintenance and convenient speed regulation, permanent magnet synchronous motors (PMSMs) are widely used in new energy vehicles, computerized numerical control (CNC) machine tools and other high-end equipment [1,2]. The proportional integral differential (PID) control technology, which is extensively utilized across various industries, governs the traditional PMSM speed regulation system. Although the PID controller has the advantages of simple structure and easy implementation [3,4], it is necessary to establish an accurate mathematical model of the controlled object to achieve accurate control. However, when modeling PMSMs, the motor structure is often simplified. This simplification has been proved inadequate when the PMSM operates at low speeds, resulting in a notable degradation of control performance, as highlighted in Ref. [5]. Moreover, noise and interference cannot be suppressed well by PID, so scholars have proposed a variety of improved PID systems [6,7]. By optimizing the adjustment parameters, the control performance can be significantly improved, but the anti-interference ability, response speed and accuracy still need to be enhanced.

Modern control theory provides a new solution for the performance improvement of control systems. Concurrently with the advancement in PMSM control technology, various effective control methods have been proposed. According to Ref. [8], the nonlinear PID controller enhances control performance through adaptive error signal transformation and dynamic adjustment of PID parameters, which enables the controller to efficiently attenuate noise in the input signal and address a trade-off between response speed and overshoot inherent in conventional PID controllers. Moreover, fuzzy PID is also adopted to enhance the system's robustness [9]. Additionally, sliding mode control (SMC) has undoubtedly emerged as a critical advancement in motor control engineering in recent years [10,11]. It can enhance the control effectiveness of PMSM systems while exhibiting rapid convergence

and strong resistance to interference. However, due to the chattering characteristics of sliding mode variable structure, how to effectively suppress this chattering has become an important topic. Robust control is also a class of methods aimed at uncertain systems, and robust control laws based on H_∞ paradigms and μ -synthesis have been studied, which achieve good disturbance suppression and robustness to parameter variations [12]. Although these algorithms effectively improve the PMSM system's control effect, they still struggle with achieving superior control performance when addressing uncertainties in models and dealing with external interferences.

Active disturbance rejection control (ADRC) is a nonlinear control strategy proposed by Jingqing Han, which can effectively deal with uncertainties and external disturbances [13]. The utility of this approach is reflected in its superior anti-interference ability, strong adaptability to the change and uncertainty of system parameters, high control precision, fast response speed, simple structure and easy implementation [14], and it has attracted wide attention in many technical fields [15–17]. ADRC consists primarily of three components: tracking differentiator (TD), extended state observer (ESO), as well as nonlinear state error feedback control law (NLSEF). The TD can smooth the input signal, reduce the noise influence and provide differential information of the signal to help the controller better respond to changes in the system. The ESO is ADRC's core component, which is used to estimate the state variables and total disturbances of the system. It has the capability to observe and mitigate uncertainties and external disturbances in real time, thereby enhancing the system's robustness and resistance to interference. Additionally, the NLSEF generates control commands based on observed state error as well as disturbance information to obtain the controlled object's accurate control. The nonlinear feedback control strategy provided by the NLSEF can dramatically enhance the system's dynamic response. ADRC inherits the essence of traditional PID and can effectively control systems with nonlinearity and uncertainty without relying on accurate models [18,19]. For example, based on ADRC and sliding mode control, Fang et al. [20] proposed a new integrated design method of speed and position loops and realized the speed and position control of PMSMs. While achieving minimal system overshoot, the complexity of parameter configuration remains a challenge. Addressing this, Li et al. [21] introduced a sliding mode ADRC, replacing the ESO with a nonlinear disturbance observer to streamline parameter adjustments. Building upon their research, Ge et al. [22] employed particle swarm optimization algorithm to enhance ADRC for spacecraft attitude control. Instead of depending upon empirical selection, the parameters are now determined via mathematical optimization. Ref. [23] proposed a fractional-order fuzzy ADRC, which mitigates the limitations of conventional control methods (such as low precision and slow response), and it is used to manage the manipulator's numerous joint motion control. Ramlavi and Chidan [24] studied the linear active disturbance rejection control tracking control approach to enhance the controller's tracking performance and robustness and solved the model uncertainty and environmental disturbance of a single-wheeled robot. Drawing on Partovibakhsh and Liu's research [25], Guo and Zhao [26] enhanced the tracking capabilities of mobile robots through the compensation mechanism of ADRC. Lv [27] proposes a fuzzy auto disturbance rejection control (Fuzzy-ADRC) method for a three-phase four-arm inverter for suppressing motor torque pulsations under complex operating conditions. Wang [28], in response to the poor control performance caused by fixed parameters in ADRC for a bearingless PMSM, proposes a dynamic parameter adjustment method for ADRC based on a genetic algorithm and backpropagation neural network. Fang [29] proposed an ADRC method based on an improved ESO to design an electromechanical actuator cascade controller for PMSMs.

Although ADRC technology effectively improves the control effect of nonlinear and uncertain systems, there are still limitations, such as the insufficiency of effective parameter-setting methodologies, the lack of estimation ability for fast time-varying disturbances and the *fal* function adopted by traditional ADRC not being smooth enough at the switching points between the nonlinear section and the linear section [30,31]. Considering the challenges currently faced in PMSM control systems with ADRC controllers, this paper aims to

make the following contribution: a polynomial nonlinear function combined with a cosine function is proposed to improve the *fal* function, which effectively improves the nonlinear function's smoothness and enhances ADRC's robustness. Investigated findings indicate that when contrasting with the conventional ADRC and other improved ADRC (IADRC), the IADRC proposed herein significantly amplifies the system's response speed, precision and robustness.

The organizational structure of this paper is outlined below. Section 2 introduces the PMSM's mathematical model. Section 3 presents ADRC's improvement strategy and constructs an IADRC controller. In Section 4, a design for the PMSM speed control system employing the proposed IADRC is presented. In Section 5, the simulation experiment and analysis are carried out, and the good control performance of the proposed IADRC for PMSM systems is verified.

2. Mathematical Modelling of PMSM

The rotor structure and permanent magnet distribution of the PMSM [32] are shown in Figure 1. According to their structural characteristics, PMSMs can be divided into two types: surface-mounted and built-in. The surface-mounted PMSM is also called the nonsalient pole PMSM. Because its permeability is very close to the vacuum permeability, the change between the reluctance and the inductance is very small, which makes the rotor have good nonsalient pole characteristics. This also makes the magnetic field of the permanent magnet have an approximately sinusoidal distribution, and the motor has a better performance. For the built-in PMSM, the reluctance of the direct axis is much higher than that of the quadrature axis, so the inductance of the direct axis is much lower than that of the quadrature axis and the rotor has salient pole characteristics, so it is also called the salient pole PMSM. The built-in PMSM realizes sensorless control through the salient pole effect and increases the power density through the electromagnetic resistance torque. However, the rotor structure is complex, the magnetic flux leakage coefficient is large and the manufacturing cost is high.

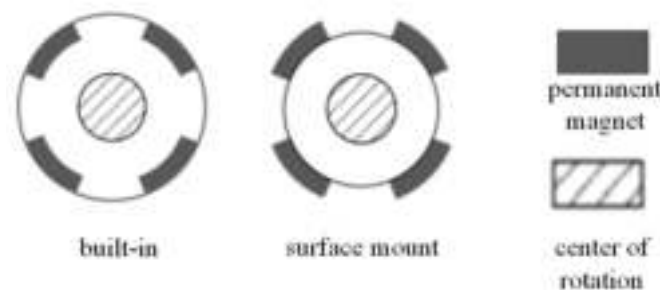


Figure 1. Schematic diagram of PMSM rotor structures.

Due to the different structures of PMSMs, their operation modes and the formations of their electromagnetic torques will also be different. Therefore, to achieve a better control effect, it is necessary to formulate a variety of different control schemes according to the internal structures of PMSMs. Considering that the surface-mounted PMSM has low cost and the same inductance of the d-q axis, it is simpler to establish the electromagnetic torque equation. Therefore, the surface-mounted PMSM is selected in this study.

To analyze the three-phase PMSM's mathematical model, the following assumptions are made for the convenience of analysis:

- (1) Ignore the reluctance of the stator and rotor cores, without considering losses due to eddy currents or hysteresis.
- (2) The permanent magnet material is nonconductive, and its permeability is equivalent to that of air.
- (3) The rotor does not have any damping windings.

- (4) The excitation magnetic field from the permanent magnet and the armature reaction magnetic field from the three-phase winding are sinusoidally distributed across the air gap.
- (5) The induced electromotive force waveform in the phase winding follows a sinusoidal pattern.

Assuming that the aforementioned conditions are met, according to electromagnetic induction and Kirchhoff's voltage law, the voltage equation of a three-phase PMSM in the natural coordinate system can be expressed as follows:

$$\begin{cases} u_a = \frac{d\psi_a}{dt} + R_s i_a \\ u_b = \frac{d\psi_b}{dt} + R_s i_b \\ u_c = \frac{d\psi_c}{dt} + R_s i_c \end{cases} \quad (1)$$

The PMSM's stator voltage equation in a two-phase rotating coordinate system can be obtained by the Clarke transform and Park transform [33]:

$$\begin{cases} u_d = \frac{d\psi_d}{dt} + R_s i_d - \omega_e \psi_q \\ u_q = \frac{d\psi_q}{dt} + R_s i_q + \omega_e \psi_d \end{cases} \quad (2)$$

The stator magnetism-chain equation is:

$$\begin{cases} \psi_d = L_d i_d + \psi_f \\ \psi_q = L_q i_q \end{cases} \quad (3)$$

Bringing Equation (3) into (2) gives:

$$\begin{cases} u_d = \frac{d\psi_d}{dt} + R_s i_d - \omega_e L_q i_q \\ u_q = \frac{d\psi_q}{dt} + R_s i_q + \omega_e (L_d i_d + \psi_f) \end{cases} \quad (4)$$

where u_a, u_b, u_c are the stator winding's three-phase voltages, with V as the unit; R_s is the stator resistance, with Ω as the unit; i_a, i_b, i_c are the stator winding's three-phase current, with A as the unit; ψ_a, ψ_b, ψ_c are the flux of the stator winding, with Wb as the unit; ψ_f is the permanent magnet flux linkage, with Wb as the unit; L_d, L_q are the inductance components of the stator inductance on the dq axis, respectively, with H as the unit; ω_e represents the electrical angular velocity of the motor, with rad/s as the unit.

Through analyzing the components of resistance torque and dynamic torque, the motor's mechanical motion equation is derived:

$$J \frac{d\omega_m}{dt} = T_e - B\omega_m - T \quad (5)$$

where J is the moment of inertia, T is the load torque and B is the viscous friction coefficient. By Clarke transformation and Park transformation, the equation of electromagnetic torque is obtained below:

$$T_e = \frac{3}{2} p_n [\psi_f i_q + (L_d - L_q) i_d i_q] \quad (6)$$

where T_e is the electromagnetic torque; $3/2$ is the coefficient when the equal amplitude transformation principle is adopted; p_n pertains to the total number of motor pole pairs.

The transformed mechanical motion equation is:

$$J \frac{d\omega_m}{dt} = \frac{3}{2} p_n [\psi_f i_q + (L_d - L_q) i_d i_q] - B\omega_m - T \quad (7)$$

3. Design of Improved ADRC Controller

The traditional ADRC treats uncertainty, internal system disturbances, and external disturbances collectively as the system's overall disturbance. And it can achieve real-

time accurate estimation and compensation of this disturbance using the ESO and NLSEF. This approach does not rely on an exact model of the plant and significantly enhances anti-interference capabilities. However, the traditional ADRC's *fal* function is not smooth enough at the switching points between the nonlinear section and the linear section, resulting in the control effect not being ideal. Therefore, an improved ADRC controller underpinned by an improved *fal* function is proposed in this paper.

3.1. Improved Fal Function Design

The *fal* function is a nonlinear function, which plays a vital role in ADRC. It not only allows ADRC to better estimate and compensate for system disturbances, thereby improving the system's anti-disturbance performance, but also allows the amplification to be mitigated when the error is large and increased when the error is small. It is beneficial to enhance the system's reaction velocity to large errors and the control accuracy to small errors, while enhancing the robustness and adaptability of the system.

In traditional ADRC, $fal(e, \alpha, \delta)$, as a crucial nonlinear function, can effectively estimate both internal and external disturbances of the system, thereby generating the ADRC's output signal. The form of the *fal* function is shown in Equation (8):

$$fal(e, \alpha, \beta) = \begin{cases} |e|^\alpha \text{sign}(e), & |e| \geq \delta \\ \frac{e}{\delta^{1-\alpha}}, & |e| < \delta \end{cases} \quad (8)$$

Taking $\delta = 0.001$, when α takes different values, the curve of the *fal* function is shown in Figure 2.

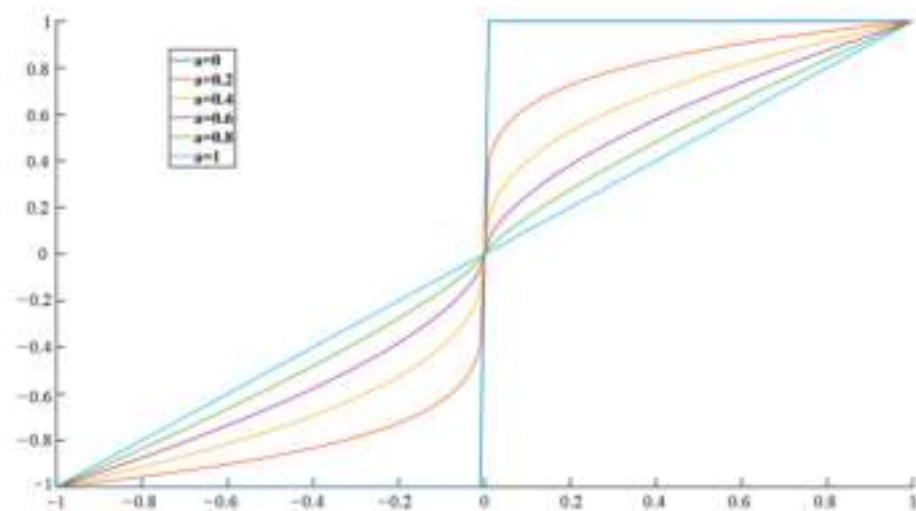


Figure 2. Curves of $fal(e, \alpha, \delta)$ function under different α values.

From the perspective of convergence, the terminal attractor $|e|^\alpha \text{sign}(e)$ is finite-time convergent, especially suitable for the control near the origin stage ($|e| < 1$), which has an amplification effect on the error, but not suitable for the control away from the origin stage, and the convergence will be slower than the classical linear control. The linear term is exponentially convergent, which is suitable for the control far from the origin ($|e| > 1$). However, in the *fal* function, the use of the terminal attractor term and the linear term is reversed, which causes the *fal* function as a whole to converge in nonfinite time, no matter how small the initial error $e(0)$ is, so the linear term can be improved to enhance the convergence ability.

From Equation (8), it is evident that the derivative values at the breakpoints differ after applying the derivative of the *fal* function. And it can also be seen from Figure 2 that the $fal(e, \alpha, \delta)$ function curves are continuous but not smooth when $e = |\delta|$, and there is a sudden change, which will lead to poor control performance of the system.

Based on the above problems, a polynomial nonlinear function $Ifal(e, \alpha, \delta)$ combining linear and trigonometric functions is proposed in this paper. The reasons for adopting a combination of trigonometric functions and polynomials are as follows:

- (1) Trigonometric functions and polynomials have explicit mathematical forms, making them easy to handle analytically and theoretically.
- (2) Properly designed trigonometric functions and polynomials can reduce system overshoot and enhance the response speed and stability of the system.
- (3) Although the added trigonometric function makes $Ifal$ more complex than the original fal function, the calculation efficiency can be improved by the look-up table and interpolation methods, and a better control effect can be achieved on the premise of ensuring real-time performance and lower hardware requirements.

The function expression is shown below:

$$Ifal(e, \alpha, \beta) = \begin{cases} |e|^\alpha \text{sign}(e) & , |e| \geq \delta \\ \frac{\delta^{\alpha-3} e^{k_1} \cos(e-\delta)}{k_1} - \frac{\delta^{\alpha-2} e^{k_2} \cos(e-\delta)}{k_2} + k_3 \delta + \alpha \delta^{\alpha-1} e & , |e| < \delta \end{cases} \quad (9)$$

The design idea of the function is as follows: considering that the main problem of the original fal function is that the segments are not smooth enough, the original linear function is changed into a polynomial function combined with a trigonometric function and linear function when $|e| \leq \delta$ in this paper. Then, according to the condition that the fal function is continuous at the segments and the derivative function values are equal, the relationship between the three constant coefficients of k_1 , k_2 and k_3 is obtained.

Based on the above ideas, it can be known that when $e = |\delta|$, the function values on both sides of the $Ifal$ function are equal, and after taking the derivative of the $Ifal$ function, the derivative values on both sides are also equal.

In light of the aforementioned concepts, the following equation is established:

$$\begin{cases} Ifal_-(e, \alpha, \beta) = Ifal_+(e, \alpha, \beta) \\ Ifal'_-(e, \alpha, \beta) = Ifal'_+(e, \alpha, \beta) \end{cases} \quad (10)$$

The derivative expression of the $Ifal$ function, after calculation, is as follows:

$$Ifal'(e, \alpha, \beta) = \begin{cases} \alpha \delta^{\alpha-1} & , |e| > \delta \\ \delta^{\alpha-3} e^{(k_1-1)} - \delta^{\alpha-2} e^{(k_2-1)} + \alpha \delta^{\alpha-1} & , |e| \leq \delta \end{cases} \quad (11)$$

Combining Equations (10) and (11), we can derive Equation (12).

$$\begin{cases} \frac{\delta^{\alpha-3} e^{k_1} \cos(e-\delta)}{k_1} - \frac{\delta^{\alpha-2} e^{k_2} \cos(e-\delta)}{k_2} + k_3 \delta + \alpha \delta^{\alpha-1} e = |e|^\alpha \text{sign}(e) \\ \delta^{\alpha-3} e^{(k_1-1)} - \delta^{\alpha-2} e^{(k_2-1)} + \alpha \delta^{\alpha-1} = \alpha \delta^{\alpha-1} \end{cases} \quad (12)$$

When $|e| = \delta$, Equation (3) simplifies to the following form:

$$\begin{cases} \frac{\delta^{k_1+\alpha-3}}{k_1} - \frac{\delta^{k_2+\alpha-2}}{k_2} + k_3 \delta + \alpha \delta = \delta^\alpha \\ \delta^{k_1+\alpha-3} - \delta^{k_2+\alpha-2} + \alpha \delta^{\alpha-1} = \alpha \delta^{\alpha-1} \end{cases} \quad (13)$$

Then, the relationship for k_1 , k_2 and k_3 is derived, $k_2 = k_1 - 1$, $k_3 = \frac{k_1^2 - k_1 + 1}{k_1(k_1 - 1)} - \alpha$, which is brought into the $Ifal$ function to obtain the new nonlinear function:

$$Ifal(e, \alpha, \delta) = \begin{cases} |e|^\alpha \text{sign}(e) & , |e| > \delta \\ \frac{\delta^{\alpha-3} e^{k_1} \cos(e-\delta)}{k_1} - \frac{\delta^{\alpha-2} e^{k_1-1} \cos(e-\delta)}{k_1-1} + \frac{\delta^{k_1+\alpha-3}}{k_1(k_1-1)} + (1-\alpha)\delta^\alpha + \alpha \delta^{\alpha-1} e & , |e| \leq \delta \end{cases} \quad (14)$$

The fal function curves before and after improvement are drawn by MATLAB 2020a from MathWorks and are shown in Figure 3.

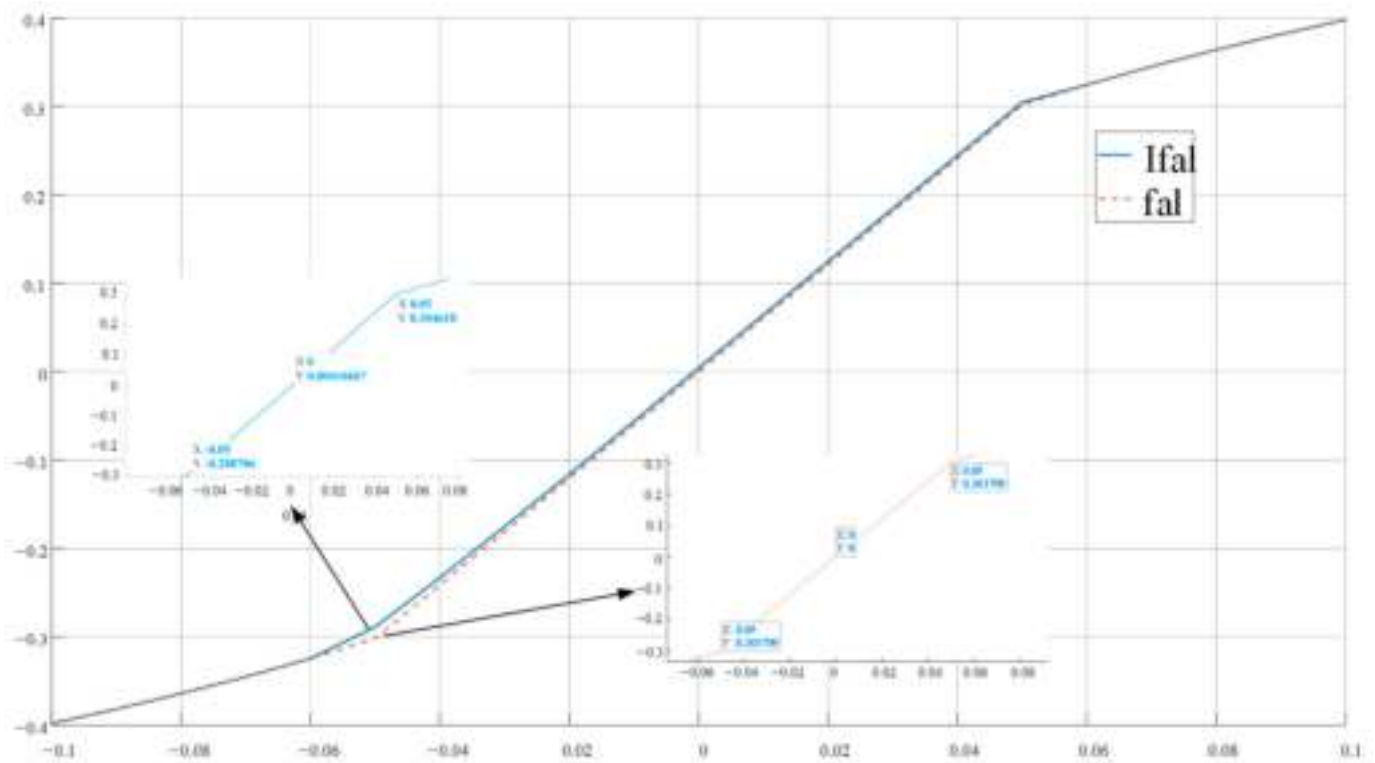


Figure 3. Curves of $fal(e, \alpha, \delta)$ function before and after improvement.

It can be seen from the graph that the $Ifal(e, \alpha, \delta)$ function curve is smoother than that of the fal function. The $Ifal$ function is continuous and derivable at the points that $|e| = \delta$. In the range of $(-0.06, 0.06)$, the output of the $Ifal$ function is higher than that of fal function, that is, the $Ifal$ function converges faster and more smoothly.

3.2. Improved ESO Design (IESO)

Moreover, the IESO serves as the nucleus for IADRC. It not only provides the state estimation of the PMSM system, but also classifies all types of both internal and external disturbances, in addition to uncertainties comprising the entirety of the PMSM system's disturbances, and estimates and compensates for the total disturbance in real time. Similar to the ESO, the IESO relies solely on the input and output data of the system and does not require an accurate mathematical model of disturbances. This approach effectively enhances the performance and robustness of the control system. The formulation of the IESO is as follows:

$$\begin{cases} e_2 = z_{21} - n \\ \dot{z}_{21} = bu(t) + z_{22} - k_{31}Ifal(e_2, \alpha_2, \delta_2) \\ \dot{z}_{22} = -k_{32}Ifal(e_2, \alpha_2, \delta_2) \end{cases} \quad (15)$$

3.3. Improved NLSEF Design (INLSEF)

The NLSEF realizes the effective control of nonlinearity, uncertainty and external disturbance in ADRC and enhances the system's control performance and robustness. The fal function is a key nonlinear element in the NLSEF for processing error signals. It can adaptively manipulate the amplification based on the magnitude of error, diminishing the gain in instances of substantial error and elevating the gain when the error is minimal, thereby enhancing the control mechanism's accuracy and robustness. The expression of the INLSEF is as follows:

$$\begin{cases} e_1 = z_{11} - z_{21} \\ u(t) = -\frac{z_{22}}{b} + k_2Ifal(e_1, \alpha_1, \delta_1) \end{cases} \quad (16)$$

where z_{21} represents an observation of the controller's dynamic response, while z_{21} denotes an observation of the disturbance, b is the compensation factor and k_2 is the regulator gain.

3.4. Design of PMSM Speed Control System Based on IADRC-Proposed

Figure 4 depicts the schematic diagram of the IADRC-proposed PMSM speed control system. The controller indirectly controls the speed and torque of the PMSM by controlling the inverter's output. The system adopts double closed-loop control, wherein the current loop operates under PI, while the speed loop is controlled by IADRC-proposed.

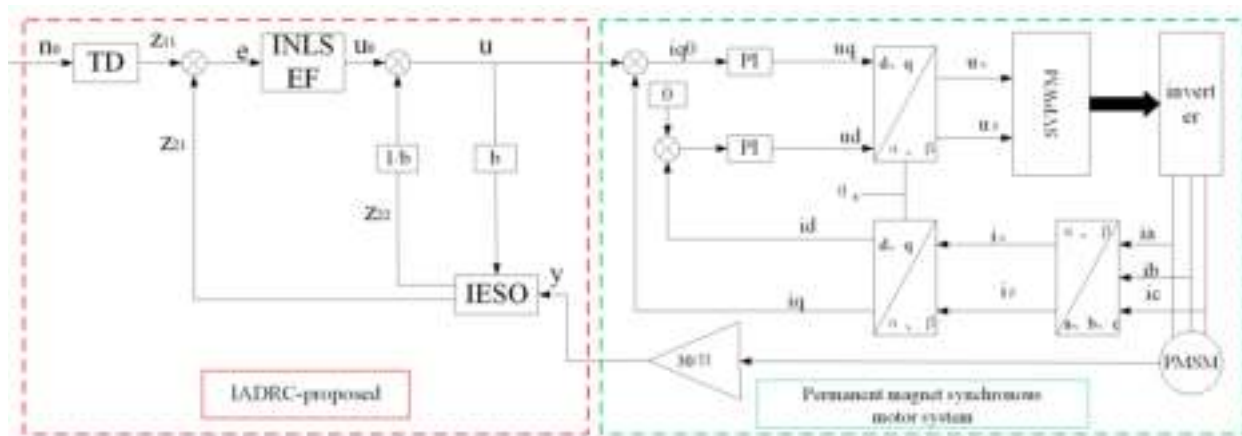


Figure 4. Diagram outlining PMSM's IADRC-proposed speed control system.

4. Simulation and Experimental Evaluation

To verify the effect of the IADRC-proposed algorithm on PMSM speed control, the traditional ADRC, SMC, fuzzy PID and the IADRC in Ref. [34] are adopted as the comparison algorithms. The PMSM speed regulation system models based on these five control algorithms are built in MATLAB/Simulink, and the simulation comparison experiments are carried out thereafter. The specifications for the PMSM in the simulation models are outlined in Table 1.

Table 1. Specifications for PMSM.

Specifications	Value
Stator phase resistance R (Ohm)	2.785
Stator direct-axis inductance L_D (H)	8.5×10^{-3}
Stator cross-axis inductance L_Q (H)	8.5×10^{-3}
Permanent magnet flux linkage ψ_f (Wb)	0.175
Inertia J ($\text{kg} \cdot \text{m}^2 / \text{rad}$)	1×10^{-3}
Pole pairs, P	4
Coefficient of friction B	1×10^{-4}

The methodology incorporated within this manuscript is delineated as follows: the expected speed of the PMSM is 1500 r/min. To verify the robustness of the IADRC-proposed algorithm, a load torque is added first. When the actual speed reaches the expected speed and is stable, the load torque of 2 N·m is suddenly applied to the PMSM's output shaft at 0.1 s to compare the no-load response curve and anti-interference ability of the PMSM under the action of the above five control algorithms. The total experimental time is 0.15 s.

The parameter settings in the IADRC-proposed algorithm are as follows. According to the specific descriptions of the ADRC's three components in Section 3, the two parameters α, δ in the TD and NLSEF are set based on experience, $\alpha_1 = 0.4, \delta_1 = 0.09, \alpha_2 = 0.4, \delta_2 = 0.09$. The two parameters in the ESO are determined through continuous simulation and tuning: $\alpha_3 = 0.25, \delta_3 = 0.04$. Velocity coefficient k_1 in the TD, regulator gain k_2 and compensation factor b of the NLSEF, derived from the mathematical model and parameters of the PMSM

system, are taken as $k_1 = 5355$, $k_2 = 5000$, $b = 1030.66$, respectively. Similarly, the calibration gains of the output in the ESO are taken as $k_{31} = 8500$, $k_{32} = 500,000$. The key parameters in the controller are shown in Table 2.

Table 2. The values of key parameters of controller.

Component	Specifications	Value
TD	Tracking factor α_1	0.4
	Filter factor δ_1	0.09
	Velocity factor k_1	5355
NLSEF	Tracking factor α_2	0.4
	Filter factor δ_2	0.09
	Regulator gain k_2	5000
	Compensation factor b	1030.66
ESO	Tracking factor α_3	0.25
	Filter factor δ_3	0.04
	The calibration gains of the output k_{31}	8500
	The calibration gains of the output k_{32}	500,000

According to the above parameter settings, the three algorithms are applied to the PMSM's speed control. The results of the experiment are shown below:

It can be seen from Figures 5 and 6 that in the PMSM's initial start-up stage, the IADRC proposed in this paper has the best effect among the three algorithms. It not only has no overshoot but also reaches and stabilizes at the expected speed at 0.027 s. The maximum error fluctuation value is 31.28 r/min, which is about 2.09% of the expected speed. The IADRC in Ref. [34] needs 0.045 s to reach and stabilize at the expected speed, and the overshoot is 61.64 r/min, which is about 4.11% of the expected speed. And the time for the traditional ADRC to reach and stabilize at the expected speed is 0.085 s, and its overshoot is also the largest, which is 119.04 r/min, about 7.89% of the expected speed. The SMC requires 0.075 s to reach and stabilize at the expected speed, with a maximum overshoot of 423.71 r/min, which is approximately 28.25% of the expected speed. The fuzzy PID takes 0.043 s to reach and stabilize at the expected speed, with a maximum error fluctuation of 30.27 r/min, which is about 2.02% of the expected speed.

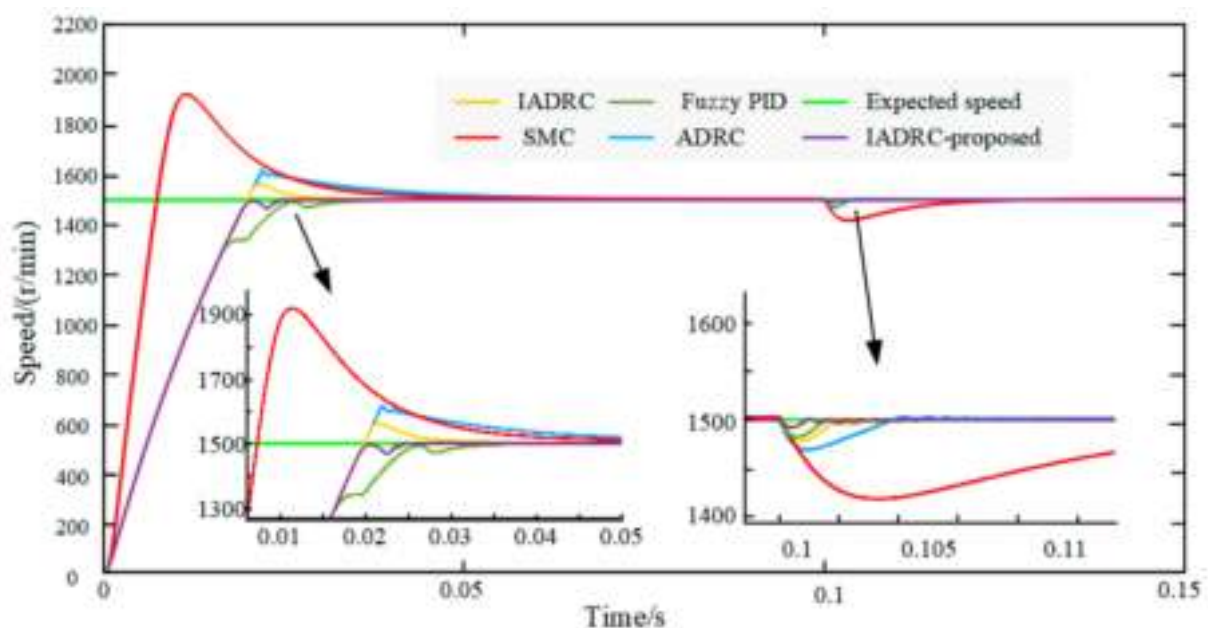


Figure 5. Speed response curves of 5 algorithms without Gaussian noise.

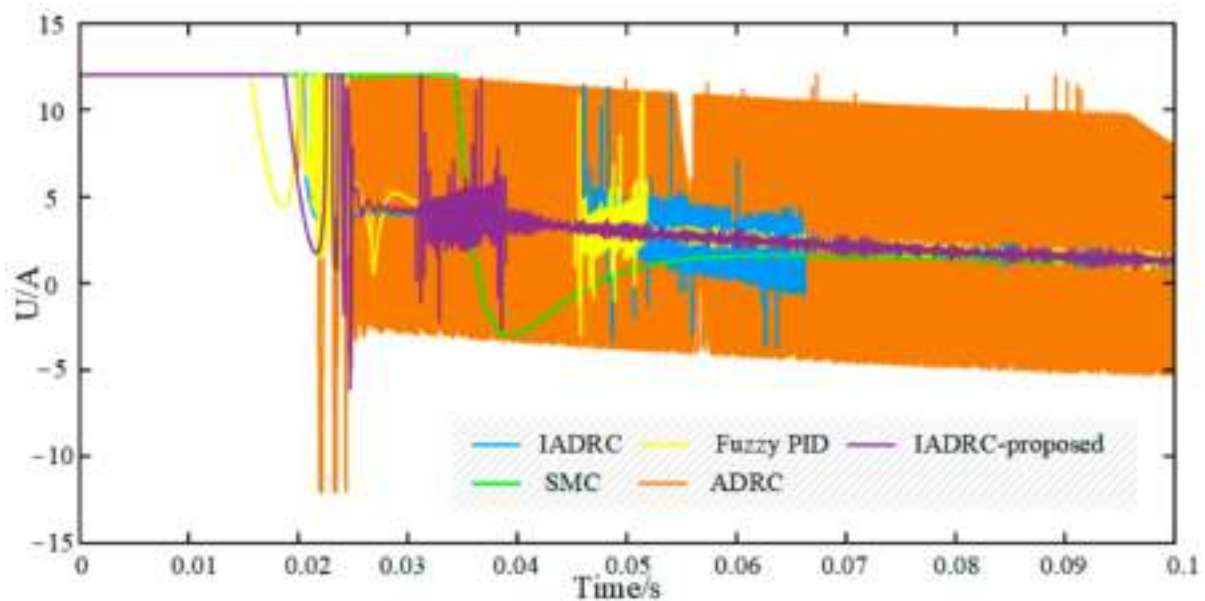


Figure 6. The plant control inputs of 5 algorithms without Gaussian noise.

When the speed of the PMSM is stabilized at the expected speed, the step load torque is applied at 0.1 s. The IADRC algorithm proposed in this paper is basically unaffected, and the speed curve only fluctuates slightly. The maximum speed error is 7.39 r/min, which is about 0.49% of the expected speed, and then quickly (0.004 s) returns to the expected value. The speed curve of the IADRC algorithm in Ref. [34] produced a maximum speed error of 20.73 r/min, which is about 1.38% of the expected speed, and returned to the expected value after 0.006 s. And the speed curve of the traditional ADRC algorithm produces a maximum speed error of 29.85 r/min, which is about 1.99% of the expected speed, and returns to the expected value after 0.034 s. The velocity curve of the SMC algorithm produces a maximum speed error of 81.98 r/min, which is approximately 5.47% of the expected speed, and returns to the expected value after 0.028 s. The velocity curve of the fuzzy PID algorithm produces a maximum speed error of 18.11 r/min, which is about 1.21% of the expected speed, and returns to the expected value after 0.07 s.

Various uncertain factors affect the PMSM system in practical work, such as environmental noise and resistance changes caused by continuous operation of the motor or changes in ambient temperature. To verify the robustness of IADRC-proposed algorithm under different parameter variations and uncertainties, simulation experiments for noise interference and motor resistance change are added to the system.

To verify the robustness of the IADRC-proposed algorithm, this paper also includes comparative experiments of the plant control input U under Gaussian noise, comparing it with other algorithms. The Gaussian noise is added at the beginning, and the signal is set to have a mean of 1 and a variance of 200. Other motor parameters are shown in Tables 1 and 2. The simulation results are shown in Figures 7 and 8.

Comparing Figures 5–8, it is obvious that the speed response curves and the system output U do not change significantly before and after the Gaussian noise is added for all five algorithms. Moreover, the speed response curves and the system output U controlled by the IADRC-proposed algorithm have the shortest oscillation time and the smallest oscillation amplitude, which verifies the stability of the IADRC-proposed algorithm.

To further validate the impact of motor resistance changes on the IADRC-proposed, the motor resistances are set to 2.5875, 2.875 and 3.1625, respectively. Other parameters are consistent with Table 1. The experimental results are shown in Figure 9.

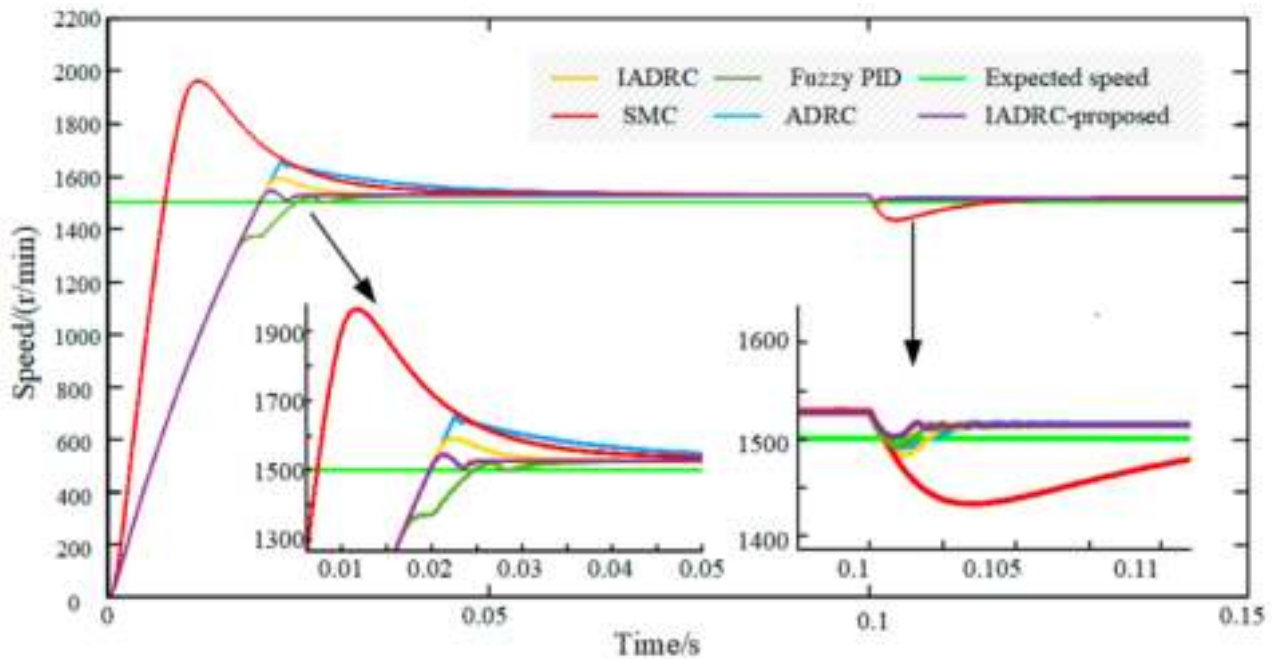


Figure 7. Speed response curves of 5 algorithms under Gaussian noise.

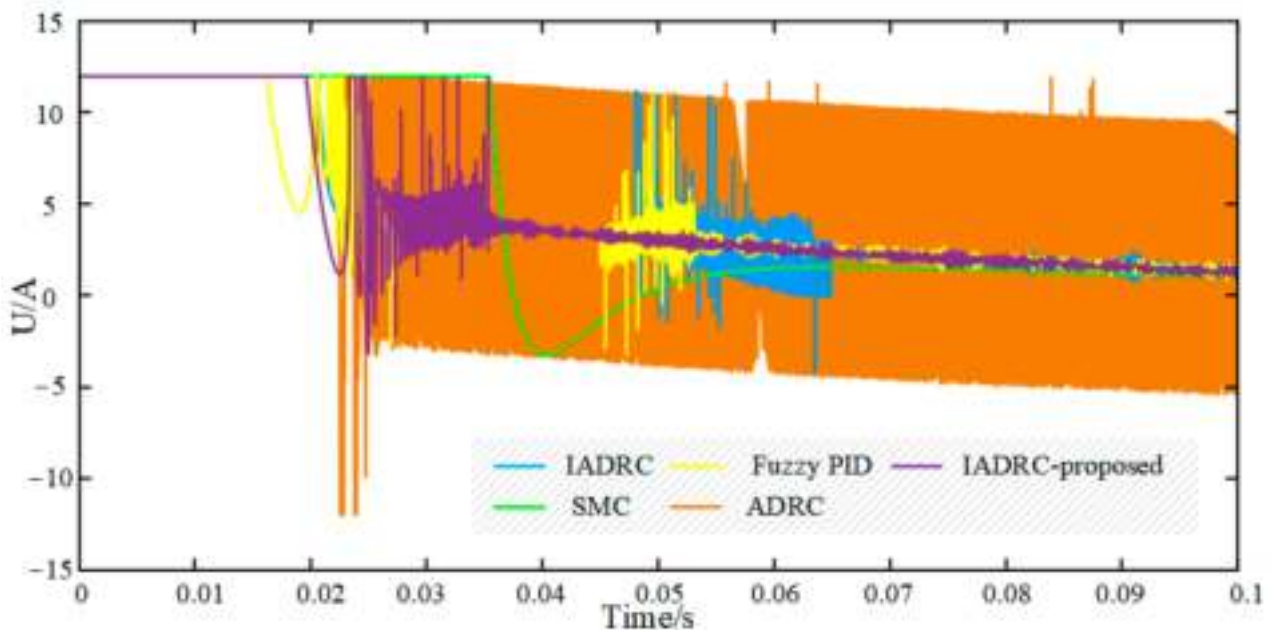


Figure 8. The plant control inputs of 5 algorithms under Gaussian noise.

From Figure 9, it is obvious that although the overshoot and the time to reach stability all change, the change is slight. It indicates that after changes in external conditions, the control effect is still very good, confirming the robustness of the IADRC-proposed algorithm.

In summary, it is effectively proved that our presented IADRC exhibits enhanced responsiveness, higher control accuracy, stronger robustness and anti-interference ability. The specific performance is as follows: our proprietary IADRC algorithm is 40.00% better than the IADRC in Ref. [34], 68.24% better than the traditional ADRC, 64.00% better than SMC and 37.21% better than fuzzy PID in terms of rapidity. In terms of accuracy, it is 49.25% better than the IADRC in Ref. [34], 73.72% better than the traditional ADRC, 92.62% better than SMC and 3.35% worse than fuzzy PID. Moreover, when the step load torque is

applied in the stable operation state, the IADRC algorithm proposed in this paper is 33.33% better than the IADRC in Ref. [34], 88.24% better than the traditional ADRC, 85.71% better than SMC and 94.29% better than fuzzy PID in terms of rapidity. In terms of accuracy, it is 64.35% better than the IADRC in Ref. [34], 75.24% better than the traditional ADRC, 90.99% better than SMC and 59.19% better than fuzzy PID. Additionally, experimental tests adding Gaussian noise and changing motor resistance attest that the proposed IADRC algorithm has good robustness.

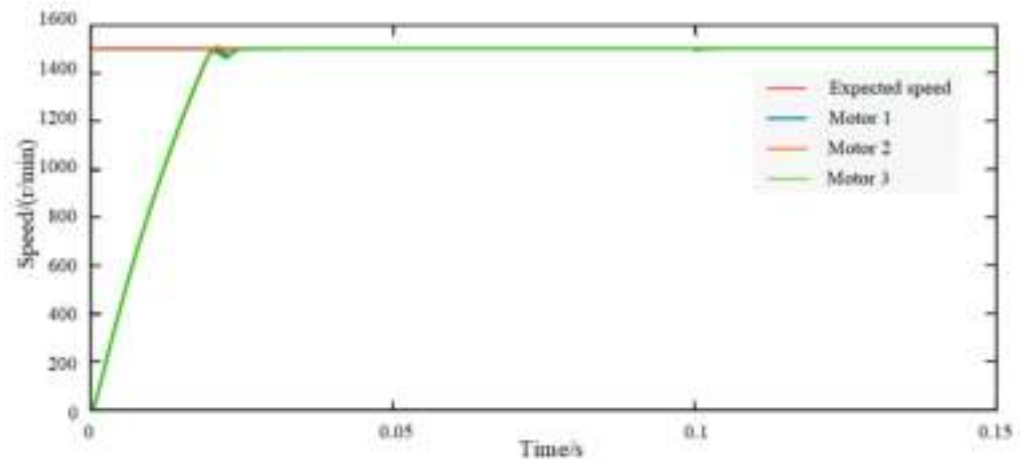


Figure 9. Motor speed curves with different resistances.

5. Conclusions

A refined *fal* function-based IADRC strategy is elucidated in this article, and a better control performance for the PMSM is obtained compared to the traditional ADRC, IADRC in Ref. [34], SMC and fuzzy PID. Simulation results indicate that compared to ADRC, IADRC in Ref. [34], SMC and fuzzy PID, the IADRC proposed in this paper has faster response speed. The time to reach and stabilize at the expected speed is 68.24% shorter than that of ADRC, 40.00% shorter than that of IADRC in Ref. [34], 64.00% shorter than that of SMC and 37.21% shorter than that of fuzzy PID. Moreover, the IADRC proposed in this paper has no overshoot, and the maximum error reduction amounts to 73.72% compared to ADRC, 49.25% compared to IADRC in Ref. [34], 92.62% compared to SMC and −3.35% compared to fuzzy PID. In the case of sudden disturbance when the speed of the PMSM is stabilized at the expected speed, the time to reach and stabilize at the expected speed is 88.24% shorter than that of ADRC, 33.33% shorter than that of IADRC in Ref. [34], 85.71% shorter than that of SMC and 94.29% shorter than that of fuzzy PID. Additionally, the maximum error of the IADRC proposed in this paper is 75.24% lower than that of ADRC, 64.35% lower than that of IADRC in Ref. [34], 90.99% lower than that of SMC and 59.19% lower than that of fuzzy PID. Apart from that, the good robustness is also verified by adding Gaussian noise and changing motor resistance.

Our future work will focus on applying the IADRC-proposed algorithm to the physical control of PMSMs and continue to improve the algorithm according to the actual control situation.

Author Contributions: Writing—original draft preparation, Z.H.; writing—review and editing, Y.C.; resources, S.C.; supervision, X.Z., J.X. and S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under grant number 52205316 and Zhejiang Provincial Natural Science Foundation of China under grant number LTGN24E050002. And the APC was funded by Wenzhou Major Science and Technology Innovation Project of China under grant number ZN2022002.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: The authors are grateful to the editors and the anonymous reviewers for their insightful comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bolognani, S.; Tubiana, L.; Zigliotto, M. Extended Kalman filter tuning in sensorless PMSM drives. *IEEE Trans. Ind. Appl.* **2003**, *39*, 1741–1747. [\[CrossRef\]](#)
2. Wallmark, O.; Harnefors, L.; Carlson, O. Control algorithms for a fault tolerant PMSM drive. *IEEE Trans. Ind. Electron.* **2007**, *54*, 1973–1980. [\[CrossRef\]](#)
3. Gashti, A.; Akbarimajd, A. Designing anti-windup PI controller for LFC of nonlinear power system combined with DSTS of nuclear power plant and HVDC link. *Electr. Eng.* **2020**, *102*, 793–809. [\[CrossRef\]](#)
4. Errouissi, R.; Al-Durra, A.; Muyeen, S.M. Experimental validation of a novel PI speed controller for AC motor drives with improved transient performances. *IEEE Trans. Control Syst. Technol.* **2018**, *26*, 1414–1421. [\[CrossRef\]](#)
5. Zaky, M.; Khater, M.; Yasin, H.; Shokralla, S. Very low speed and zero speed estimations of sensorless induction motor drives. *Electr. Power Syst. Res.* **2010**, *80*, 143–151. [\[CrossRef\]](#)
6. Shi, R. Improvement of predictive control algorithm based on fuzzy fractional order PID. *J. Intell. Syst.* **2023**, *32*, 876. [\[CrossRef\]](#)
7. Tan, W.; Han, W.; Xu, J. State-Space PID: A Missing Link Between Classical and Modern Control. *IEEE Access* **2022**, *10*, 116540–116553. [\[CrossRef\]](#)
8. Atalik, G.; Senturk, S. Intuitionistic fuzzy control charts based on intuitionistic fuzzy ranking method for TIFNs. *Soft Comput.* **2022**, *26*, 11403–11407. [\[CrossRef\]](#)
9. Tu, W.W.; Dong, J.X. Robust sliding mode control for a class of nonlinear systems through dual-layer sliding mode scheme. *J. Frankl. Inst. Eng. Appl. Math.* **2023**, *360*, 10227–10250. [\[CrossRef\]](#)
10. Zhang, G.C.; Li, X.F.; Xia, Y.Q.; He, S. Adaptive sliding mode control for 2D nonlinear Fornasini-Marchesini model subject to quantisation and packet dropouts. *Int. J. Syst. Sci.* **2021**, *52*, 3001–3012. [\[CrossRef\]](#)
11. Zhao, J.; Zhao, T.; Liu, N. Fractional-Order Active Disturbance Rejection Control with Fuzzy Self-Tuning for Precision Stabilized Platform. *Entropy* **2022**, *24*, 1681. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Xiang, C.D.; Petersen, I.R.; Dong, D.Y. Coherent robust H_∞ control of uncertain linear quantum systems with direct and indirect couplings. *J. Frankl. Inst. Eng. Appl. Math.* **2023**, *360*, 13845–13869. [\[CrossRef\]](#)
13. Han, J. From PID to active disturbance rejection control. *Trans. Ind. Electron.* **2009**, *56*, 900–906. [\[CrossRef\]](#)
14. Liu, J.J.; Chen, H.; Wang, L.; Yu, M.Y. Active disturbance rejection control for improved depth model of AUV. *Appl. Mech. Mater.* **2014**, *687*, 157–162. [\[CrossRef\]](#)
15. Meng, Y.; Liu, B.; Wang, L. Speed control of PMSM based on an optimized ADRC controller. *Math. Probl. Eng.* **2019**, *2019*, 18. [\[CrossRef\]](#)
16. Xiong, S.; Xie, H.; Song, K.; Zhang, G. A speed tracking method for autonomous driving via ADRC with extended state observer. *Appl. Sci.* **2019**, *9*, 3339. [\[CrossRef\]](#)
17. Wan, J.; Liu, H.; Yuan, J.; Shen, Y.; Zhang, H.; Wang, H.; Zheng, Y. Motion Control of Autonomous Underwater Vehicle Based on Fractional Calculus Active Disturbance Rejection. *J. Mar. Sci. Eng.* **2021**, *9*, 1306. [\[CrossRef\]](#)
18. Yang, Z.; Qian, C.; Sun, X.; Wang, K. Enhanced Linear ADRC Strategy for Sensorless Control of IPMSM Considering Cross-Coupling Factors. *IEEE Trans. Transp. Electr.* **2023**, *9*, 4437–4446. [\[CrossRef\]](#)
19. Zuo, Y.F.; Zhang, J.; Liu, C.; Zhang, T. Integrated design for permanent magnet synchronous motor servo systems based on active disturbance rejection control. *Trans. China Electrotech. Soc.* **2016**, *31*, 51–58. [\[CrossRef\]](#)
20. Fang, S.; Meng, J.; Meng, Y.; Wang, Y.; Huang, D. Discrete-Time Active Disturbance Rejection Current Control of PM Motor at Low Speed Using Resonant Sliding Mode. *IEEE Trans. Transp. Electr.* **2023**, *9*, 4783–4794. [\[CrossRef\]](#)
21. Li, Z.; Zhang, Z.; Wang, J.; Wang, S.; Chen, X.; Sun, H. ADRC Control System of PMLSM Based on Novel Non-Singular Terminal Sliding Mode Observer. *Energies* **2022**, *15*, 3720. [\[CrossRef\]](#)
22. Ge, X.; Sun, K. Optimal control of a spacecraft with deployable solar arrays using particle swarm optimization algorithm. *Sci. China Technol. Sci.* **2011**, *54*, 1107–1112. [\[CrossRef\]](#)
23. Lamraoui, H.C.; Qidan, Z. Speed tracking control of unicycle type mobile robot based on LADRC. In Proceedings of the 2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE), Beijing, China, 17–19 August 2017; pp. 200–204. [\[CrossRef\]](#)
24. Partovibakhsh, M.; Liu, G. Slip ratio estimation and control of wheeled mobile robot on different terrains. In Proceedings of the 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Shenyang, China, 8–12 June 2015; pp. 566–571. [\[CrossRef\]](#)
25. Ding, L.; Gao, H.-B.; Deng, Z.-Q.; Li, Z.; Xia, K.-R.; Duan, G.-R. Path-following control of wheeled planetary exploration robots moving on deformable rough terrain. *Sci. World J.* **2014**, *2014*, 793526. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Feng, H.; Guo, B.-Z. Active disturbance rejection control: Old and new results. *Annu. Rev. Control* **2017**, *44*, 238–248. [\[CrossRef\]](#)

27. Lv, C.; Wang, B.; Chen, J.; Zhang, R.; Dong, H.; Wan, S. Research on a Torque Ripple Suppression Method of Fuzzy Active Disturbance Rejection Control for a Permanent Magnet Synchronous Motor. *Electronics* **2024**, *13*, 1280. [[CrossRef](#)]
28. Wang, X.; Zhu, H. Active Disturbance Rejection Control of Bearingless Permanent Magnet Synchronous Motor Based on Genetic Algorithm and Neural Network Parameters Dynamic Adjustment Method. *Electronics* **2023**, *12*, 1455. [[CrossRef](#)]
29. Fang, Q.; Zhou, Y.; Ma, S.; Zhang, C.; Wang, Y.; Huangfu, H. Electromechanical Actuator Servo Control Technology Based on Active Disturbance Rejection Control. *Electronics* **2023**, *12*, 1934. [[CrossRef](#)]
30. Sun, C.; Liu, M.; Liu, C.; Feng, X.; Wu, H. An Industrial Quadrotor UAV Control Method Based on Fuzzy Adaptive Linear Active Disturbance Rejection Control. *Electronics* **2021**, *10*, 376. [[CrossRef](#)]
31. Štumberger, B.; Štumberger, G.; Hadžiselimović, M.; Hamler, A.; Goričan, V.; Jesenik, M.; Trlep, M. Comparison of torque capability of three-phase permanent magnet synchronous motors with different permanent magnet arrangement. *J. Magn. Magn. Mater.* **2007**, *316*, e261–e264. [[CrossRef](#)]
32. Zhang, X.G.; Fang, S.J.; Zhang, H. Predictive Current Error Compensation-Based Strong Robust Model Predictive Control for PMSM Drive Systems. *IEEE Trans. Ind. Electron.* **2024**, 1–10. [[CrossRef](#)]
33. Vitor, A.L.O.; Scalassara, P.R.; Goedtel, A.; Endo, W. Patterns Based on Clarke and Park Transforms of Wavelet Coefficients for Classification of Electrical Machine Faults. *J. Control Autom. Electr. Syst.* **2023**, *34*, 230–245. [[CrossRef](#)]
34. Xu, P. Research on PMSM Speed Control System of Mining Motor Vehicle Based on Fuzzy Self-Immunity. Master's Thesis, Anhui University of Technology, Anhui, China, 2022.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Investigating and Mitigating the Performance–Fairness Tradeoff via Protected-Category Sampling

Gideon Popoola and John Sheppard * 

Gianforte School of Computing, Montana State University, Bozeman, MT 59717, USA;
gideon.popoola@student.montana.edu

* Correspondence: john.sheppard@montana.edu

Abstract: Machine learning algorithms have become common in everyday decision making, and decision-assistance systems are ubiquitous in our everyday lives. Hence, research on the prevention and mitigation of potential bias and unfairness of the predictions made by these algorithms has been increasing in recent years. Most research on fairness and bias mitigation in machine learning often treats each protected variable separately, but in reality, it is possible for one person to belong to multiple protected categories. Hence, in this work, combining a set of protected variables and generating new columns that separate these protected variables into many subcategories was examined. These new subcategories tend to be extremely imbalanced, so bias mitigation was approached as an imbalanced classification problem. Specifically, four new custom sampling methods were developed and investigated to sample these new subcategories. These new sampling methods are referred to as protected-category oversampling, protected-category proportional sampling, protected-category Synthetic Minority Oversampling Technique (PC-SMOTE), and protected-category Adaptive Synthetic Sampling (PC-ADASYN). These sampling methods modify the existing sampling method by focusing their sampling on the new subcategories rather than the class label. The impact of these sampling strategies was then evaluated based on classical performance and fairness in classification settings. Classification performance was measured using accuracy and F1 based on training univariate decision trees, and fairness was measured using equalized odd differences and statistical parity. To evaluate the impact of fairness versus performance, these measures were evaluated against decision tree depth. The results show that the proposed methods were able to determine optimal points, whereby fairness was increased without decreasing performance, thus mitigating any potential performance–fairness tradeoff.

Keywords: fairness; protected categories; machine learning; sampling



Citation: Popoola, G.; Sheppard, J. Investigating and Mitigating the Performance–Fairness Tradeoff via Protected-Category Sampling. *Electronics* **2024**, *13*, 3024. <https://doi.org/10.3390/electronics13153024>

Academic Editors: Niussha Shafiabady and Jianlong Zhou

Received: 11 June 2024

Revised: 23 July 2024

Accepted: 24 July 2024

Published: 31 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As machine learning (ML) algorithms increasingly dominate decision-making and decision-assistance systems, their widespread deployment across various sectors raises pressing issues about the fairness and transparency of their predictions [1]. The potential for these algorithms to perpetuate or exacerbate existing societal biases has propelled a significant body of research to investigate and mitigate algorithmic unfairness. This is critical because the decisions influenced by these algorithms profoundly impact individuals, affecting outcomes in domains ranging from finance and employment to criminal justice and healthcare [2].

The source of unfairness and bias in ML is multifaceted [3]. In particular, it is possible that unfairness arises directly from the ML algorithms themselves due to a possible misalignment of the underlying *inductive bias* of the algorithms vis-à-vis the target concept and data distribution. This is referred to as *algorithmic bias*. An alternative concern lies in potential bias resident in the data used to train the models where, as a direct result of the typical “independent and identically distributed” (IID) assumption employed in most ML

methods, the result of learning is to propagate the bias in predictions such that they match the bias in the underlying data itself. It is this latter situation that constitutes the focus of our work here.

1.1. Bias and Unfairness in Machine Learning

Bias is the prejudicial, unfair, or unequal treatment of an individual or group based on specific features, often referred to as sensitive or protected features [4]. Examples of these protected features include age, race, disability, sex, and gender [5]. Bias in ML can be divided roughly into disparate treatment (direct unfairness) and impact treatment (indirect unfairness) [6]. Direct unfairness happens when protected features are used explicitly in making decisions. Indirect unfairness has become increasingly common today. This type of unfairness does not use protected attributes explicitly; instead, it occurs when reliance on variables associated with these attributes results in significantly different outcomes for the protected groups. These other variables are known as proxy features. Examples of real-world bias include the historical U.S. practice of “redlining”, where home mortgages were denied to residents of zip codes predominantly inhabited by minorities, Amazon hiring process gender bias, Google soap dispenser racial bias, etc. [7].

Though these decision assistance tools help automate the decision-making process, these tools may result in unfair treatment of either individuals or groups, both directly or indirectly [8]. Unfairness can occur in several areas of modeling, such as in the training dataset. This can happen when the training dataset does not provide a fair representation of the protected categories, so the “ground truth” becomes difficult to determine. For example, consider a dataset from a company where a specific group has historically faced discrimination. Specifically, suppose female employees in this company have not been promoted as their male counterparts, who, in contrast, have seen career advancement, despite both groups performing at the same level. In this situation, the true value of female employee contributions—the ground truth—is not visible. As a result, an ML algorithm trained on this data is likely to detect and incorporate this bias, thereby perpetuating existing prejudices. This could lead to the algorithm making discriminatory decisions, such as recommending male candidates for hire or promotion more frequently than equally or more qualified female candidates.

Another area where unfairness can occur is in the ML algorithm itself [9]. ML algorithms can still produce discriminatory decisions, even when trained on an unbiased dataset where the “ground truth” is represented accurately. This situation arises when the system’s errors disproportionately impact individuals from a specific group or minority. For example, consider a breast cancer detection algorithm that exhibits significantly higher false negative rates for Black individuals compared with White individuals, meaning it fails to identify breast cancer more frequently in Black patients than in White patients. If this algorithm is used to inform treatment recommendations, it would erroneously advise against treatment for a greater number of Black individuals than White, leading to racial disparities in healthcare outcomes. This underscores the critical need to ensure that algorithms perform equitably across all groups in terms of their training data and how their errors affect different populations. Results from previous literature have reported several cases of algorithms resulting in unfair treatment, e.g., redlining and racial profiling [10], mortgage discrimination [11], employment and personnel selection [12].

While considerable efforts have been geared toward addressing bias in ML predictions [13,14], much of the existing research has focused on mitigating bias for single protected attributes in isolation [15]. For example, on a dataset with two protected attributes, race, and sex, most existing approaches can learn either a fair model involving race or a fair model involving sex but not a fair model involving both race and sex [7]. However, real-world identities are not singular; they are complex and multifaceted, with individuals often belonging to multiple protected groups simultaneously [16]. For example, an individual can be discriminated against across several protected attributes such as age, race, and sex simultaneously. This intersectionality can lead to compounded forms of bias and

discrimination, which are not adequately addressed by single-variable fairness interventions. Therefore, it is critical to develop methodologies that holistically address personal identities' multidimensional nature. This project seeks to bridge this gap by considering combinations of protected categories, thereby synthesizing these protected categories into comprehensive multicategory groups, and aims to tackle the layered complexities of bias more effectively using novel protected-category sampling methods, thus acknowledging and addressing the multifaceted nature of personal identities and potential biases.

The work presented in this paper is motivated by the problem of using ML algorithms for decision making in socially sensitive areas such as loan assessment, hiring, or mortgage assessment, working with this situation where an individual can belong to several protected categories. Given a labeled training dataset containing two or more protected features, the method proposed combines these protected attributes and then splits them into new multicategories. These new categories are likely to be extremely imbalanced and need to be balanced to improve the fairness of the prediction of our ML algorithms. Popular sampling methods such as over-sampling [17], Synthetic Minority Oversampling Technique (SMOTE) [18], Adaptive Synthetic Sampling (ADASYN) [19], etc., sample data across class labels, which does not align with the goal of our research of sampling across the new multicategories. Hence, a new class of modifications of these sampling methods is proposed that can sample across the new category rather than class labels. This new class of modified sampling is called *protected-category sampling*. The resulting proposed protected-category sampling methods are used to sample and balance the new categories before performing classification. The novelty of this work is two-fold. First, the proposed approach combines the protected categories to form new multicategories that mimic what the identity human being looks like in the real world. The second is the modification of existing sampling methods to conform with the sampling of these new categories in order to make sure that all the new categories have the same number of instances.

For demonstration purposes only, a univariate decision tree was chosen as the classification algorithm. The intent is to demonstrate the effects of the different sampling methods on performance, expecting that similar trends will be exhibited regardless of the underlying learning method. The proposed sampling method was compared with the baseline (unsampled data) using accuracy and F1 as the classification performance metrics, as well as equalized odds differences and statistical parity as the fairness metrics. Also, several analyses were performed to show how maximum depth in the decision tree affects both accuracy and fairness.

1.2. Research Question

Proceeding from empirical observation that a trade-off sometimes exists between fairness and ML performance [20], this research tries to answer several questions, such as how this trade-off might be mitigated. In particular, we seek to answer whether the protected-category sampling method of tackling fairness can mitigate this trade-off. In addition, can we develop a methodological framework that effectively mitigates biases across these combined protected variables without compromising the predictive accuracy of ML models? Finally, we plan to answer the question of how the depth of a decision tree affects both accuracy and fairness metrics, thus exploring the relationship between the level of fit (underfitting through overfitting) and fairness.

1.3. Hypothesis

We hypothesize that, by employing sophisticated protected-category sampling techniques designed for these newly formulated multicategory groups, we can significantly increase model fairness in terms of equalized odds differences without decreasing classification performance in terms of accuracy and F1. Furthermore, we explore the delicate balance between fairness and accuracy, hypothesizing that it is possible to identify strategic points where fairness can be maximized without detrimental impacts on performance. This research challenges existing claims of the existence of trade-offs in fairness and ML

prediction. It sets the stage for future explorations into the multidimensional nature of identity and discrimination in automated decision systems.

1.4. Contributions

The broad problem of fairness in machine learning is significant in that the prevalence of AI and ML systems today is having a major impact on people's lives and livelihoods. While attention to fair ML has increased substantially, there continues to be a need for methods to advance fair ML without negatively impacting ML performance. Based on an in-depth review of the literature and the above need for this type of work, the methods reported here make the following contributions:

1. The commonly-held assumption that there exists an inherent tradeoff between fairness and performance (i.e., accuracy) in machine learning is challenged with evidence provided to support this challenge. In particular, the results in this paper indicate that such a tradeoff can be mitigated, suggesting that any tradeoff is most likely tied to how the data is being managed.
2. Four novel preprocessing methods for sampling data are presented based on applying a multicategory sampling strategy using data captured in protected categories. The methods proceed from the assumption and corresponding hypothesis that balancing the data based on these multicategory properties can increase fairness without adversely affecting machine learning model performance.
3. Experimental results are presented using three datasets studied extensively within the fair ML community. The experiments include comparisons with traditional methods of training with no resampling to demonstrate the relative effects of the proposed methods. The results demonstrate that two of the proposed methods, Protected-Category Synthetic Minority Oversampling Technique (PC-SMOTE) and Protected-Category Adaptive Synthetic sampling (PC-ADASYN), are particularly effective in improving both fairness and performance.
4. A detailed analysis relating the potential effects of underfitting and overfitting on fairness is presented by examining different levels in a decision tree model, with and without using the proposed sampling methods. The results demonstrate the ability of the proposed methods to identify an ideal level of the tree where both fairness and accuracy are maximized.

As a result of the above contributions, this work represents a significant step forward in addressing concerns of fairness in machine learning. A key takeaway from the methods and results reported here is that fairness can be addressed without compromising model performance.

1.5. Organization

This paper is organized as follows. In Section 2, a detailed explanation of fairness and a discussion of several technical fairness metrics are presented. Then, in Section 3, previous literature related to bias mitigation strategies is described. In Section 4, we describe our proposed sampling techniques, dataset, and approach to hyperparameter tuning. In Section 5, we present the results of several experiments along with statistical hypothesis tests as a means of validating these results. In Section 6, the experimental results are discussed, and how each algorithm performs on each dataset and each metric is analyzed. Further results on the impact of tree depth on fairness and accuracy are presented as well. In Section 7, the limitations of this work and corresponding directions for future work are presented, and Section 8 presents a number of conclusions.

2. Background

This study considers fairness when predicting an outcome $y \in \mathcal{Y}$ from a set of features $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ and some additional protected attributes $\mathbf{s} \in \mathcal{S} \subseteq \mathbb{R}^p$, such as race, gender, and sex. For example, in loan prediction, $\mathbf{x}$ represents an applicant's financial history, $\mathbf{s}$ is their self-reported race and gender, and y is whether their loan is approved

or denied. A prediction model is considered fair if its errors are evenly distributed across protected groups like different races or genders. The class predictions from training data $\mathcal{D}$ are denoted as $\hat{Y}_{\mathcal{D}} := h(\mathbf{x}, \mathbf{s})$ for some $h : \mathcal{X} \times \mathcal{S} \rightarrow \mathcal{Y}$ from a class $\mathbf{H}$. The protected attributes $\mathbf{s} \in \mathcal{S}$ in our study are assumed to be binary with a special value n denoting the unprivileged group. For example, $\mathcal{S}$ could be race and n “non-White”; therefore, the binary nature of $\mathcal{S}$ is $\{w, n\}$ where w represents White applicants, who are the privileged group, and n represents non-White applicants, who are the unprivileged group. The definition can be further generalized to nonbinary cases.

Discrimination in labeled datasets can be defined as given a dataset $\mathcal{D}$, feature set $\mathcal{X}$, and protected attribute set $\mathcal{S}$ with domain value $\{w, n\}$. The discrimination in $\mathcal{D}$ with respect to the group $\mathcal{S} = n$ denoted as $dis_{s=n}(\mathcal{D})$ is defined as

$$dis_{s=n}(\mathcal{D}) = \frac{|\{\mathbf{x} \in \mathcal{D} : \mathbf{x}(s) = w, h(\mathbf{x}) = +\}|}{|\{\mathbf{x} \in \mathcal{D} : \mathbf{x}(s) = w\}|} - \frac{|\{\mathbf{x} \in \mathcal{D} : \mathbf{x}(s) = n, h(\mathbf{x}) = +\}|}{|\{\mathbf{x} \in \mathcal{D} : \mathbf{x}(s) = n\}|}$$

The above definition can be translated to the difference in the probability of an applicant being in the positive class for each protected attributes domain $\{w, n\}$. Our study extends the above definition by considering dataset $\mathcal{D}$, which contains two or more protected attributes.

Two popular fairness metrics are used. The first is *equalized odd difference* (EOD), which measures how discriminative or fair our prediction is. EOD states that a binary classifier $\hat{y}$ is fair if its false negative rate (FNR) and true positive rate (TPR) are equal across the domain of $\mathcal{S}$ [21]. FPR and TPR with respect to protected attribute $\mathbf{s} \in \mathcal{S}$ with value n can be defined as

$$TPR_n(\hat{y}) = P(\hat{y} = 1 | y = 1, S = n)$$

$$FPR_n(\hat{y}) = P(\hat{y} = 1 | y = 0, S = n)$$

EOD is then defined mathematically as the difference between TPR and FPR across different groups in a protected attribute. That is,

$$EOD = TPR_n(\hat{y}) - FPR_n(\hat{y}).$$

A fair classifier has an EOD of 0, while an unfair classifier has an EOD of 1. Although achieving a fully fair classifier in practice is almost impossible, this research is geared toward improving EOD without decreasing accuracy. Then, for EOD,

$$FPR_n(\hat{y}) = P(\hat{y} = 1 | y = 0, S = n) = TPR_n(\hat{y}) = P(\hat{y} = 1 | y = 1, S = n)$$

and

$$FPR_w(\hat{y}) = P(\hat{y} = 1 | y = 0, S = w) = TPR_w(\hat{y}) = P(\hat{y} = 1 | y = 0, S = w)$$

To extend the above EOD definition to our multicategory, the EOD is calculated for each column, then the macroaverage of the EOD is presented as the final EOD. The second metric used to measure fairness in ML prediction is *statistical parity* (SP). SP defines fairness as an equal probability of being classified as positive [22]. This can be interpreted as each group in a protected attribute having the same probability of being classified with a positive outcome.

$$P(\hat{y} = 1 | S = w) = P(\hat{y} = 1 | S = n)$$

3. Literature Review

ML algorithms, increasingly utilized for decision making in critical applications such as recidivism, credit scoring, loan decisions, etc., might initially be assumed to be fair and free of inherent bias. However, in reality, they may inherit any bias or discrimination present in the data on which they are trained, as noted by Burt [23]. Moreover, merely removing protected variables from the dataset is insufficient to tackle indirect discrimination and

might, in fact, conceal it. This recognition has heightened the need for more advanced tools, making discovering and preventing discrimination a significant area of research, as highlighted by [24–27].

Bias in ML is a fast-growing topic in the machine learning research community. Bias in an ML model can lead to an unfair treatment of people belonging to certain protected groups. Lately, industrial leaders have started putting more and more emphasis on bias in ML models and software. The Institute for Electrical and Electronics Engineers (IEEE) [28], Microsoft [29], and the European Union [30] have recently published principles for guiding fair AI conduct. These organizations have stated that ML models must be fair in real-world applications. Bias mitigation strategies involve modifying one or more of the following to ensure the predictions made by the ML algorithm are less biased: (a) the training data, (b) the ML algorithm, and (c) the ensuing predictions themselves. These are, respectively, categorized as preprocessing [31], inprocessing [32], and postprocessing approaches [21].

First, the training data can be preprocessed to lower unfairness or bias before training the model. Kamiran and Calders [6] suggest sampling or reweighting the data to neutralize discrimination. This approach can adjust the representation or importance of certain data points to favor (or reduce favor) one class over another. Another method involves changing individual data records directly to reduce discrimination, as explored by [33]. For example, this approach involves altering values in a dataset to decrease identifiable biases against certain groups. Additionally, the concept of *t*-closeness, introduced by Sondeck et al. [34], is applied to discrimination control in the work of [35]. Using *t*-closeness ensures that the distribution of sensitive attributes in any given group is close to the distribution of the attribute in the entire dataset, thereby preserving privacy and preventing discrimination based on sensitive attributes. A common thread among these approaches is balancing discrimination control with the processed data's utility, that is minimizing bias without significantly compromising the data's accuracy, representativeness, and overall usefulness for predictive modeling or analysis. This balance is essential for ensuring that efforts to promote fairness do not inadvertently reduce the quality or applicability of the data.

Overall, the pre-processing method can further be divided into three categories: (1) data modification, (2) data removal, and (3) data resampling. Methods in the first category aim to modify the values of the training data points (including protected attribute values, class values, and feature values) to lower the bias in the dataset. An example of this method is data massaging proposed by [15]. Their approach ranks the training data, and data close to the decision boundary in both privileged and unprivileged groups are flipped. Alternatively, an optimized pre-processing method that learns a probabilistic transformation that edits the classes and features with individual distortion and group fairness was proposed by Fahse et al. [23]. In [36], the original attribute values are replaced with values chosen independently from the class label to train a model roughly achieving equalized odds. Similarly, Peng et al. [37] replace the protected attribute values with values predicted based on other attributes, similar to data imputation.

Methods in the second category aim to train a fair model by removing certain features from the training set. An example of this method is data suppression proposed by Dhar et al. [38]. In their paper, the protected attributes and features that are highly correlated with protected attributes, otherwise known as proxy attributes, are removed from the dataset to train a fair model.

Methods in the third category aim to train a fair model either by adjusting the sample weights or by oversampling the dataset. For example, Krasanakis et al. [39] proposed a reweighting method that iteratively adapts training sample weights with a theoretically grounded model to mitigate the bias–accuracy tradeoff. In [40], Chakraborty et al. proposed FairSMOTE as a method to over-sample training points from minority groups with artificial data points based on Synthetic Minority Oversampling Technique (SMOTE) [18], to achieve balanced class distributions. Also, Yan et al. [41] proposed oversampling the training data from the minority groups with artificial data points to achieve balanced class distribu-

tions. Unlike FairSMOTE, the authors focused on scenarios where protected attributes are unknown and applied a clustering method to identify different demographic groups.

Inprocessing involves methods that modify the way an ML model is trained as a means to reduce bias. In [42], an adversarial debiasing approach was proposed. This approach learns a classifier to increase accuracy and fairness in prediction by including a variable for the group interested by simultaneously learning a predictor and an adversary. This leads to the generation of an unbiased classifier because the predictions do not contain any group discrimination information that the adversary could utilize. Alternatively, an algorithm that takes a fairness metric as part of the loss function and returns a model trained for that fairness metric was proposed in [43]. Kamishima et al. [22] proposed a regularization method, which included a penalty term in the loss function of a classifier to produce an unbiased prediction. Zafar et al. [44] developed a new weighting method whereby they tune the sample weight for each training datum to achieve a specific fairness objective, such as equalized odds on the validation data. Recently, bias mitigation has been approached as a constrained optimization problem by adding a fairness constraint and optimizing the loss to be consistent with that constraint [45,46]. Also, some works modify neural networks by using dropout to drop neurons that belong to protected attributes [47].

Postprocessing methods mitigate bias after fitting an ML model and include approaches such as calibration, constraint optimization, and transformation thresholding [6]. Such methods propose an algorithm that gives favorable outcomes to unprivileged groups and unfavorable outcomes to favorable groups within a given confidence interval around the decision boundary with the highest uncertainty. For example, one approach modifies the peak thresholds of the classifier to yield a specified equal opportunity or equalized odds target. Yet another approach involves randomly mutating the classes of certain predictions into different classes [48].

Several new studies [49,50] combined either preprocessing, inprocessing, or postprocessing to form an ensemble method. For example, Bhaskaruni et al. [50] combine oversampling the imbalance protected class with a decision boundary shifting a postprocessing method to tackle the unfairness problem.

Researchers have delved into various concepts of discrimination and fairness within algorithmic decision making. Disparate impact (referred to previously as indirect fairness), for example, is measured through statistical parity and group fairness, as discussed by Bhaskaruni et al. [50]. On the other hand, the concept of individual fairness, also introduced by Bhaskaruni et al., emphasizes that similar individuals should be treated similarly, regardless of their group affiliation. This approach focuses on fairness at the individual level, ensuring that decisions are made based on relevant attributes rather than group-based stereotypes or biases.

In classifiers and other predictive models, achieving equal error rates across different groups is a key goal, as highlighted by Zhang and Neill [16]. Similarly, ensuring calibration or the absence of predictive bias in the predictions, as discussed by Hardt et al. [21], is crucial. However, the tension between these notions—calibration and equal error rates—is explored by Dwork et al. [51] and Pleiss et al. [52], indicating that simultaneously satisfying both can be challenging. Karimi-Haghighi and Castillo [53] present related work exploring the complexities inherent in achieving algorithmic fairness. Friedler et al. [54] further examines the trade-offs in meeting various algorithmic fairness definitions, especially from a public safety perspective. Given that our work focuses on preprocessing rather than modeling, considerations such as balanced error rates and predictive bias become less directly applicable.

Based on our review of various preprocessing methods, it appears that no work has been conducted attempting to model fairness for two or more protected attributes simultaneously. Also, the sampling method used in prior work focused only on sampling based on class labels rather than the protected categories. Hence, in this paper, preprocessing is emphasized as it represents the most adaptable aspect of the data science pipeline [55]. Preprocessing is distinct in that it does not depend on the choice of modeling algorithm and can be seamlessly

incorporated with data release and publishing mechanisms. This independence and flexibility make preprocessing critical for ensuring data quality and fairness before any analytical or predictive modeling occurs. Finally, we focus on new custom sampling methods that sample the protected category in the data training to build a fair model.

4. Methodology

The focus of our work is to explore sampling methods to enhance fairness in ML without the corresponding prediction performance suffering, thus mitigating the fairness–performance tradeoff. As a result, Four novel sampling methods focused on achieving this goal are proposed. These sampling methods address the imbalanced class problem posed by the new multicategory generated due to the combination of the protected categories. Custom sampling methods are needed because the existing methods sample data based on minority and majority classes, but to mitigate fairness, the new multicategories are sampled to be equal. This, in turn, calls for modifying the existing sampling methods to sample data based on these new categories. This leads to four new sampling methods: protected-category oversampling, protected-category proportional sampling, protected-category SMOTE (PC-SMOTE), and protected-category ADASYN (PC-ADASYN).

4.1. Protected-Category Oversampling

In protected-category oversampling, the first step is to combine the protected categories in the dataset and encode the combination to produce our new multicategory. For example, in the Adult Income dataset, age (young and adult), race (White and others), and sex (male and female) are combined to generate eight new categories, which become ADULTWHITEMALE, ADULTOTHERSMALE, YOUNGWHITEMALE, YOUNGOTHERSMAIL, ADULTWHITEFEMALE, ADULTOTHERSFEMALE, YOUNGWHITEFEMALE, and YOUNGOTHERSFEMALE, respectively. These new categories have varying sample sizes, and the goal of our protected-category oversampling is to balance this new category such that the sample size of each of the new categories matches the size of the category with the highest sample size. To avoid data leakage, the dataset is separated into train and test, applying oversampling only on the training data and then testing on an unsampled test set.

The pseudocode in Algorithm 1 shows our protected category oversampling method in detail. In the algorithm, the largest category was used as the baseline because it is the category with the highest sample size. The sampling process results in new training data with a balanced sample size across the new category. The algorithm works by sampling the rest of the protected categories to match the sample size of the baseline. This sampling is performed by repeating the categories multiple times along with their class labels.

4.2. Protected-Category Proportional Sampling

The Protected-Category Proportional Sampling method is a generalization of protected-category oversampling because the process begins by setting a target sample size (which is a hyperparameter to be tuned, rather than just the size of the largest multicategory), denoted as *targetSamples*. This corresponds to the desired number of instances needed for each category. This target ensures uniformity across all categories, mitigating the risk of model bias towards more frequent categories. The typical result of applying this method is that some categories that have more samples than the *targetSamples* will be under-sampled while others will be oversampled to yield an equal proportion of them in the training dataset. The pseudocode in Algorithm 2 shows the step-by-step of the protected-category proportional sampling method.

Algorithm 1 Protected-Category Oversampling

```

1: baselineCount  $\leftarrow$  sum of entries in 'Largest_Category' of  $X_{train}$ 
2: totalCount  $\leftarrow$  number of entries in  $X_{train}$ 
3: baselineProportion  $\leftarrow$  baselineCount / totalCount
4: balancedData  $\leftarrow$  initialize an empty dataset
5: categories  $\leftarrow$  list of column names in  $X_{train}$  starting with 'combined_category_'
6: for each category in categories do
7:   categoryData  $\leftarrow$  select entries in  $X_{train}$  where category = 1
8:   categoryData  $\leftarrow$  combine categoryData with corresponding labels from  $y_{train}$ 
9:   categoryCount  $\leftarrow$  number of entries in categoryData
10:  targetCount  $\leftarrow$  integer part of totalCount  $\times$  baselineProportion
11:  if categoryCount < targetCount then
12:    sampledData  $\leftarrow$  sample targetCount from categoryData with replacement
13:    balancedData  $\leftarrow$  append sampledData to balancedData
14:  else
15:    balancedData  $\leftarrow$  append categoryData to balancedData
16:  end if
17: end for
18: return balancedData

```

Algorithm 2 Protected-Category Proportional Sampling

```

1: targetSamples  $\leftarrow$  5000
2: sampledBalanced  $\leftarrow$  initialize an empty data set
3: for each column in new_categories.columns do
4:   categoryRows  $\leftarrow$  select rows in new_categories where column = 1
5:   sampledRows  $\leftarrow$  sample targetSamples entries from categoryRows with replacement
6:   for each col in oneHotEncodedBalanced.columns do
7:     sampledRows[col]  $\leftarrow$  0
8:   end for
9:   sampledRows[column]  $\leftarrow$  1
10:  sampledBalanced  $\leftarrow$  append sampledRows to sampledBalanced
11: end for
12: return sampledBalanced

```

4.3. Protected-Category SMOTE

The Protected-Category Synthetic Minority Oversampling Technique (PC-SMOTE) sampling method is a more complex process aimed at mimicking SMOTE but modified for sampling our new categories, rather than class labels. In this approach, the first step was to modify SMOTE to use a fixed number of neighbors and to randomly select one neighbor for the interpolation rather than averaging all of them. The pseudocode in Algorithm 3 shows the procedure for the PC-SMOTE. Since the intent for the method is to use it for the new category sampling, it does not address the generation of class labels directly. Hence, a new function that can generate a new class label for the synthetic data is needed. For this, a new function is defined that generates class labels based on the number of new synthetic data generated and a preselected balance ratio between the two classes. Algorithm 4 shows how our new function generates labels for our synthetic samples. The algorithm first determines the number of samples for each class based on the balance ratio and generates the sample needed for each class. The class labels are the shuffle to prevent algorithmic bias in the classes generated.

These two algorithms are combined together to form PC-SMOTE, as shown in Algorithm 5. In the approach to achieve multicategory balance, each distinct category is iterated over such that the subset of data associated with that category is identified. The number of synthetic samples needed to reach a predefined maximum size per category is then calculated. If additional samples are required, the data is generated using PC-SMOTE,

which interpolates between existing data points and their nearest neighbors. Concurrently, a balanced distribution of synthetic class labels is created with a specified balance ratio by employing Algorithm 4. These synthetic features and labels are then incorporated into the training subset for each category. The process is repeated for all categories, resulting in a balanced dataset. The hyperparameters in this Algorithm 5 are the number of neighbors and balance ratio.

Algorithm 3 Custom Synthetic Minority Oversampling Technique (SMOTE)

```

1: procedure CUSTOMSMOTE(data, n_samples)
2:   syntheticSamples  $\leftarrow$  zero matrix of size (n_samples  $\times$  number of columns in data)
3:   nn  $\leftarrow$  NearestNeighbors(n_neighbors = 7).fit(data)
4:   neighbors  $\leftarrow$  nn.kneighbors(data, return_distance = False)
5:   for i  $\leftarrow$  1 to n_samples do
6:     sampleIdx  $\leftarrow$  random integer from 0 to (number of rows in data − 1)
7:     nnIdx  $\leftarrow$  random choice from neighbors[sampleIdx, 1 :]
8:     diff  $\leftarrow$  data[nnIdx] − data[sampleIdx]
9:     weight  $\leftarrow$  random number from uniform distribution between 0 and 1
10:    syntheticSamples[i]  $\leftarrow$  data[sampleIdx] + weight  $\times$  diff
11:  end for
12:  return syntheticSamples
13: end procedure

```

Algorithm 4 Generate Balanced Synthetic Labels

```

1: procedure GENBALSYNTHLABELS(n_samplesNeeded, balanceRatio)
2:   nClass1  $\leftarrow$  int(n_samplesNeeded  $\times$  balanceRatio)
3:   nClass0  $\leftarrow$  n_samplesNeeded − nClass1
4:   syntheticLabels  $\leftarrow$  [0]  $\times$  nClass0 + [1]  $\times$  nClass1
5:   SHUFFLE(syntheticLabels) ▷ Randomly shuffle the labels
6:   return syntheticLabels
7: end procedure

```

Algorithm 5 Protected-category SMOTE

```

1: balancedDataList  $\leftarrow$  initialize an empty list
2: for each category in categories do
3:   categorySubset  $\leftarrow$  select rows in train_data s.t. 'combined_category' == category
4:   features  $\leftarrow$  remove 'class', 'combined_category' from categorySubset
5:   nSamples  $\leftarrow$  max_size − number of rows in categorySubset
6:   if nSamples > 0 then
7:     syntheticFeatures  $\leftarrow$  PCSmote(features, nSamples)
8:     syntheticLabels  $\leftarrow$  GenBalSynthLabels(nSamples, balanceRatio)
9:     syntheticFeatures['class']  $\leftarrow$  syntheticLabels
10:    syntheticFeatures['combined_category']  $\leftarrow$  category
11:    categorySubsetBalanced  $\leftarrow$  concatenate categorySubset and syntheticFeatures
12:  else
13:    categorySubsetBalanced  $\leftarrow$  categorySubset
14:  end if
15:  append categorySubsetBalanced to balancedDataList
16: end for
17: balancedData  $\leftarrow$  append balancedDataList and reset index

```

4.4. Protected-Category ADASYN

The protected-category ADASYN method mimics adaptive synthetic minority (ADASYN) sampling but is modified slightly to fulfill our goal of protected-category sampling. Our PC-ADASYN algorithm is shown in Algorithm 6. It extends ADASYN by focusing on category

density rather than class imbalance. Specifically, this function operates by finding the nearest neighbors to the data and then calculating the density of each data point's category within its immediate neighborhood. It weights these densities inversely to prioritize minority categories, making it more likely to generate synthetic samples from underrepresented categories. The synthetic samples are created by interpolating between selected data points and their neighbors, similar to SMOTE but using a random weight to vary the interpolation, thus ensuring a diverse synthetic dataset. This approach helps address the imbalance at the category level and enriches the dataset's variance, potentially improving the robustness and fairness of ML models trained on this data. Since this sampling method also generates new samples by interpolating, Algorithm 4 is used to generate class labels for the new synthetic samples.

Algorithm 6 PC-ADASYN for Category-Based Balancing

```

1: procedure PCADASYNCATEGORIES(data, labels, n_samplesNeeded, n_neighbors)
2:   n_neighbors  $\leftarrow$  n_neighbors + 1                                ▷ Including the data point itself
3:   nn  $\leftarrow$  NearestNeighbors(n_neighbors).fit(data)
4:   distances, indices  $\leftarrow$  nn.kneighbors(data)
5:   densities  $\leftarrow$  zero array of length(data)
6:   for i  $\leftarrow$  0 to length(data) − 1 do
7:     current_category  $\leftarrow$  labels[i]
8:     neighbor_indices  $\leftarrow$  indices[i][1 :]                          ▷ Skip the self index
9:     densities[i]  $\leftarrow$  SUM(labels[neighbor_indices] == current_category)
10:  end for
11:  weights  $\leftarrow$  1/(densities + 1)                                ▷ Add 1 to prevent division by zero
12:  weights  $\leftarrow$  weights/SUM(weights)                                ▷ Normalize weights
13:  syntheticSamples  $\leftarrow$  empty list
14:  sampleIndices  $\leftarrow$  random choice with replacement from length(data) using weights
15:  for each idx in sampleIndices do
16:    baseIdx  $\leftarrow$  idx
17:    neighborIdx  $\leftarrow$  RANDOMCHOICE(indices[baseIdx][1 :])
18:    diff  $\leftarrow$  data[neighborIdx] − data[baseIdx]
19:    syntheticSample  $\leftarrow$  data[baseIdx] + RANDOM()  $\times$  diff
20:    append syntheticSample to syntheticSamples
21:  end for
22:  return array(syntheticSamples)
23: end procedure

```

Algorithms 4 and 6 are combined to form our protected-category ADASYN sampling, as shown in Algorithm 7. For each category, the data corresponding to that category is isolated and the size deficit relative to the largest category is computed. If additional samples are needed, the PC-ADASYN method is applied, generating synthetic features that respect the category's distribution characteristics. These features are then complemented with synthetically generated labels, maintaining a predefined class balance ratio. The process not only corrects category imbalances but also enriches the dataset, potentially enhancing the predictive accuracy and fairness of models trained on this data.

4.5. Dataset and Hyperparameter Tuning

To test the four sampling methods, a classifier was needed to assess the effects on fairness and performance. Ultimately, the type of classifier is not directly relevant since the goal is to mitigate the fairness–performance tradeoff, rather than to find the best classifier. Therefore, we chose to use univariate decision trees based on CART [56] due to their robustness against noise and missing data. The specific implementation we chose was taken from the sklearn library (version 1.5.1) [57]. In addition, decision trees allow us to control the strength of fit by setting the tree depth of the learned tree. This allows us to compare fairness and performance across different levels of fitting.

The process begins by combining protected categories within each dataset, applying one-hot encoding to create new multicategory features, and then performing label encoding. The datasets were then divided using a stratified 10-fold cross-validation to ensure a representative distribution of classes in each fold. For each fold, training was conducted on sampled data using the previously described methods, while classification was tested on the corresponding unsampled test sets. Consistency in model training was maintained by applying identical tree depth across all sampling methods, and the results provided are averages of the 10-fold runs with their corresponding confidence intervals.

Algorithm 7 Protected-category ADASYN

```

1: balancedDataList ← initialize an empty list
2: for each category in categories do
3:   categorySubset ← select from new_data3 s.t. 'combined_category' == category
4:   features ← remove 'class', 'combined_category' from categorySubset
5:   categoryLabels ← extract 'combined_category' from categorySubset
6:   nSamplesNeeded ← max_size minus number of rows in categorySubset
7:   if nSamplesNeeded > 0 then
8:     syntheticFeatures ← PCAdasyn(features, categoryLabels, nSamplesNeeded)
9:     syntheticLabels ← GenBalSynthLabels(nSamplesNeeded, balanceRatio)
10:    syntheticFeatures['class'] ← syntheticLabels
11:    syntheticFeatures['combined_category'] ← category
12:    categorySubsetBalanced ← concatenate categorySubset with syntheticFeatures
13:  else
14:    categorySubsetBalanced ← categorySubset
15:  end if
16:  append categorySubsetBalanced to balancedDataList
17: end for
18: balancedData ← concatenate balancedDataList and reset index
  
```

Three datasets were selected from the UCI repository [58] for our analysis: the Adult Income dataset [59], the German Credit [60] dataset, and the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) dataset. The Adult Income dataset aims to predict whether an individual earns above USD 50,000, featuring eight categorical and four numerical attributes, with protected variables corresponding to age (young or adult), sex (male or female), and race (White or others). The adult income dataset was donated to UCI in 1996. The German Credit dataset, used to predict creditworthiness, comprises 20 categorical and two numerical attributes, with protected variables of age and sex. German credit dataset was donated to UCI in 1994. The COMPAS dataset [61], which assesses recidivism rates in the United States, includes six categorical and six numerical features, with protected variables of age, race, and sex. The dataset was published in 2018. These datasets were selected because they represent the state-of-the-art datasets for measuring bias and discrimination and are widely used in other studies on algorithmic bias and fairness (see Section 3). Also, the datasets have various sizes, ranging from small to large, which makes them suitable for testing our sampling methods.

Hyperparameter tuning was conducted using grid search [62] to explore a broad range of parameters, complemented by visual assessments to identify optimal settings that balance Equalized Odds Difference (EOD) and accuracy. For the Adult Income dataset, the optimal hyperparameters included a maximum tree depth of 3 and, for PC-SMOTE and PC-ADASYN, a nearest neighbor setting of 5 with a balanced ratio of 0.34. These parameters were similarly effective for the German Credit dataset. For the COMPAS dataset, a maximum tree depth of 2 was optimal for all sampling methods. PC-SMOTE and PC-ADASYN were adjusted to the nearest neighbor setting of 3 and a balanced ratio of 0.60.

5. Results

The four sampling strategies were applied to the three datasets described above and evaluated their impact using a simple univariate decision tree classifier. The results in Tables 1–3 show notable differences in model performance across five sampling strategies: no sampling, oversampling, proportional sampling, PC-SMOTE, and PC-ADASYN on our three datasets. Each method was assessed based on accuracy, macro F1, Equalized Odds Difference (EOD), and Statistical Parity (SP). The results were measured in accuracy and macro F1 because these two metrics are the most popular classification metrics. Also, limiting the metric to two makes the results comparable for statistical analysis.

To ascertain the statistical significance of each method's results, we used the Friedman test, a nonparametric alternative to the one-way ANOVA with repeated measures. Upon finding significant results from the Friedman test, we proceeded with the Nemenyi post hoc test. This test is used to evaluate pairwise comparisons between the methods to ascertain which methods statistically differ from each other. The Nemenyi test is advantageous in this setting because it accounts for multiple comparisons without assuming normal distributions, thereby providing a robust way to understand specific pairwise differences.

For the Adult Income dataset (Table 1), the no-sampling method yielded an accuracy of 0.82, setting a high baseline for comparison. However, it demonstrated a slightly biased prediction with an EOD of 0.36 and minimal disparity in prediction rates ($SP = 0.02$). In contrast, oversampling maintained the same accuracy but lowered the macro F1 slightly to 0.65, indicating potential overfitting issues while worsening fairness ($EOD = 0.66$) and increasing disparity in prediction rates ($SP = 0.09$). Proportional sampling decreased accuracy to 0.79 but improved the macro F1 to 0.79, suggesting a better balance between precision and recall. However, it significantly increased SP to 0.71, indicating a substantial disparity in positive prediction rates, which raises concerns about the model's fairness. The two custom approaches for SMOTE and ADASYN were designed specifically to improve upon these metrics. PC-SMOTE showed a moderate performance with an accuracy of 0.81 and an improved EOD of 0.25, suggesting enhanced fairness over basic oversampling and the no-sampling method. However, it still recorded lower macro F1 (0.63), indicating misrepresentation issues in synthetic data generation. PC-ADASYN proved to be the most balanced approach, maintaining high accuracy (0.82) and better handling of class imbalances, with a moderate improvement in fairness ($EOD = 0.28$) and a controlled increase in prediction rate disparity ($SP = 0.09$). Overall, the baseline accuracy is statistically significantly better than the proportional sampling while it is not statistically significant as compared with the other sampling methods. For the macro F1 proportional sampling is statistically significantly better than other sampling methods. For the EOD, PC-ADASYN is statistically better than other sampling methods while the no-sampling method SP is statistically significantly better than other sampling methods.

Table 1. Results of the sampling methods on the Adult Income dataset with 95% confidence intervals.

Sampling Method	Accuracy	Macro F1	EOD	SP
No sampling	0.82 ± 0.00	0.66 ± 0.01	0.36 ± 0.18	0.02 ± 0.00
Over-sample	0.82 ± 0.02	0.65 ± 0.06	0.66 ± 0.2	0.09 ± 0.02
Prop. Sample	0.79 ± 0.05	0.79 ± 0.06	0.46 ± 0.12	0.71 ± 0.10
PC-SMOTE	0.81 ± 0.05	0.63 ± 0.07	0.25 ± 0.21	0.07 ± 0.00
PC-ADASYN	0.82 ± 0.04	0.64 ± 0.07	0.28 ± 0.19	0.09 ± 0.02

The results of the experiments on the German Credit dataset also show varying impacts of each sampling strategy, particularly regarding fairness and accuracy, as shown in Table 2. Without sampling, the baseline model achieved an accuracy of 0.72 but exhibited significant bias in its prediction, with an EOD of 0.88, indicating a substantial disparity in error rates between groups in the protected attributes. Implementing oversampling maintained accuracy while improving the macro F1 to 0.68 and notably reducing EOD to 0.37, albeit at

the cost of increased SP to 0.35, highlighting a potential trade-off between different fairness measures. Proportional sampling reduced accuracy slightly to 0.68 but achieved the best F1-score of 0.69. It also lowered EOD to 0.32, suggesting it effectively balances prediction quality with fairness. PC-SMOTE shows an improvement in accuracy with 0.73 but a lower macro F1 of 0.55; the model's fairness shows a huge improvement over the baseline with an EOD of 0.15 and a moderate SP of 0.1. PC-ADASYN shows a similar accuracy to the baseline at 0.72, albeit with the lowest macro F1 of 0.48, suggesting a potential trade-off in precision and recall. However, the model exhibits the best in fairness prediction with an EOD of 0.13 and SP of 0.06. Overall, the result shows that the accuracy of no sampling is not statistically significant to other sampling methods except proportional samplings while for the EOD the results of all the sampling methods are statistically significant in comparison with the no-sampling method.

Table 2. Results of the sampling methods on the German Credit dataset with 95% confidence intervals.

Sampling Method	Accuracy	Macro F1	EOD	SP
No sampling	0.72 ± 0.03	0.62 ± 0.06	0.88 ± 0.10	0.07 ± 0.01
Over-sample	0.72 ± 0.05	0.68 ± 0.04	0.37 ± 0.16	0.35 ± 0.11
Prop. Sample	0.69 ± 0.04	0.69 ± 0.07	0.32 ± 0.26	0.29 ± 0.10
PC-SMOTE	0.73 ± 0.05	0.55 ± 0.09	0.15 ± 0.09	0.1 ± 0.02
PC-ADASYN	0.72 ± 0.03	0.48 ± 0.11	0.13 ± 0.02	0.06 ± 0.00

Table 3 shows the results of our experience with the COMPAS dataset. These results reveal significant variations in model performance across the different sampling strategies. The baseline approach, without sampling, achieved accuracy and a macro F1-score of 0.89 but showed higher disparities in fairness metrics, with an Equalized Odds Difference (EOD) of 0.39 and a Statistical Parity (SP) of 0.29. This underscores potential biases that unadjusted models may exhibit towards protected groups. The application of oversampling slightly improved accuracy to 0.90 but also improved fairness notably, decreasing EOD to 0.26. This suggests effectiveness in reducing outcome disparities without compromising SP. Conversely, proportional sampling, while boosting accuracy and macro F1 to 0.90 and 0.91, respectively, also achieved an EOD of 0.26, improving it over the baseline while also recording a higher SP of 0.36, indicating a potential increase in disparity of positive outcomes across groups. PC-SMOTE and PC-ADASYN, with identical scores in accuracy, macro F1, and SP, managed to maintain fairness improvements with an EOD of 0.30, though these methods also increased SP to 0.47. Overall, the results show that both the accuracy and the EOD of our sampling methods are statistically significantly better than the no-sampling method.

Table 3. Results of the sampling methods on the COMPAS dataset with 95% confidence intervals.

Sampling Method	Accuracy	Macro F1	EOD	SP
No sampling	0.89 ± 0.04	0.89 ± 0.04	0.39 ± 0.15	0.29 ± 0.17
Oversample	0.90 ± 0.03	0.90 ± 0.05	0.26 ± 0.14	0.25 ± 0.07
Prop. Sample	0.90 ± 0.05	0.91 ± 0.04	0.26 ± 0.12	0.36 ± 0.10
PC-SMOTE	0.91 ± 0.02	0.91 ± 0.02	0.30 ± 0.11	0.47 ± 0.19
PC-ADASYN	0.91 ± 0.02	0.91 ± 0.03	0.30 ± 0.13	0.47 ± 0.21

6. Discussion

Results of the experiments on the three datasets substantiate that protected-category sampling can markedly enhance model fairness, often without significantly compromising prediction accuracy. In some cases, improvement in accuracy and macro F1 were also demonstrated. Focusing on the Adult Income dataset results, PC-SMOTE and PC-ADASYN

demonstrated notable improvements in EOD and maintained moderate levels of SP. The efficacy of these methods can largely be attributed to their sophisticated interpolation techniques. For example, a visual examination of the decision trees generated with no sampling and PC-ADASYN provides insightful contrasts. Examples from a single representative fold are shown in Figures 1 and 2, respectively. The decision tree learned without sampling selected its root with a feature closely associated with protected attributes, thus acting as a proxy attribute. This led to pronounced prediction bias as reflected in the EOD. Conversely, the decision tree trained on data generated using PC-ADASYN began with a feature that generalized predictions very well and mitigated bias, as evidenced by a notable enhancement in model fairness and a higher Gini impurity, indicating a purer initial split.



Figure 1. Example decision tree trained on Adult Income with no sampling.

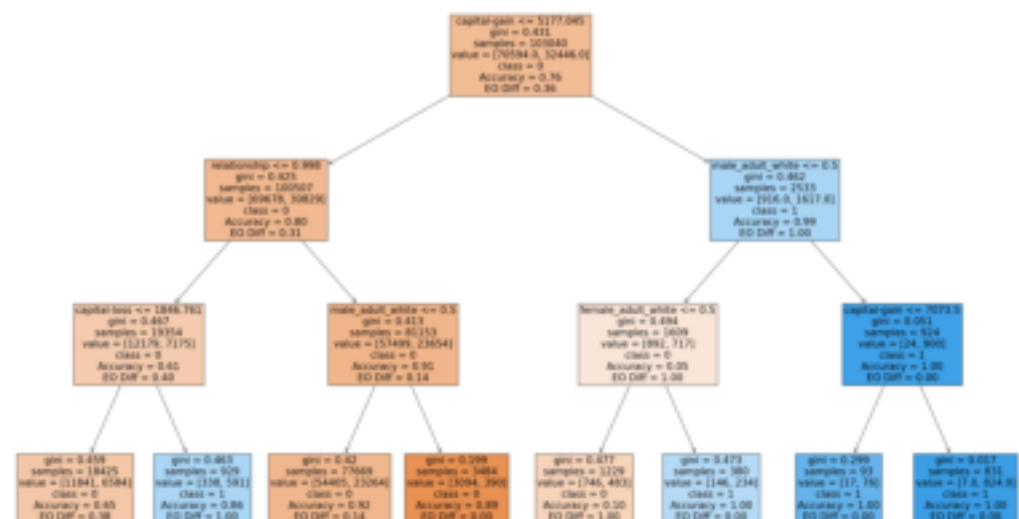


Figure 2. Decision tree of PC-ADASYN on adult income.

6.1. Comparing Fairness vs. Performance

Comparing oversampling and proportional sampling, the methods' approaches to augmenting sample size by duplicating existing data rows were straightforward and did not yield substantial improvements in EOD. This outcome makes sense since these methods tend to replicate existing biases, which can potentially exacerbate fairness issues rather than alleviate them. This is evident, in particular, when considering the Adult Income dataset, where the classes are extremely imbalanced. These naïve replication strategies lack the interpolation capacity of PC-SMOTE and PC-ADASYN to adjust samples near decision boundaries, which is crucial for mitigating the bias in the dataset. In contrast, the interpolation strategies used by SMOTE and ADASYN expand the dataset and enhance its diversity. This is particularly effective for samples near decision boundaries, where slight shifts in the features can affect the fairness of predictions significantly. By interpolating between samples, SMOTE and ADASYN effectively move these boundary samples towards more equitable regions of the feature space, thus directly confronting and reducing bias more effectively than methods that increase sample volume without altering data structure. The class generation function (Algorithm 4) also helps increase the overall class distribution. The results of our PC-SMOTE and PC-ADASYN on Adult income also show superiority over the results obtained in [41], where the accuracy of 0.59 and SP of 0.17 was obtained. Also, the results of PC-SMOTE and PC-ADASYN show superiority over the results obtained in [40], where an EOD of 0.89 and a slightly better accuracy of 0.84.

In examining the results on the German Credit dataset, we observed a trend similar to what was noted in the Adult Income dataset: the no-sampling method has a very high bias regarding EOD. The low SP of 0.07 indicates minimal disparity in the positive prediction rates between the groups, but this in itself is not a good way of measuring fairness since the favored group has more samples than the unfavored one. This has the effect of skewing the calculation of SP since it only counts positive decisions in each group, which are influenced by sample size. One takeaway is the importance of employing multiple fairness metrics to view a model's impact on all stakeholders comprehensively. For oversampling, we saw an improvement in EOD with a similar SP; this shows that increasing the number of samples for each of the multicategory's protected attributes improves the fairness with respect to EOD. In addition, the updated SP reflects what it will look like to have a more equal number of samples for each multicategory, unlike in the baseline where the favored group has five times more samples than the unfavored group. The accuracy of proportional sampling drops because the baseline number of samples selected after hyperparameter tuning was insufficient for the model to generalize the unsampled test set, leading to overfitting. The overfitting was confirmed by considering the training accuracy. Interestingly, the model is not trading accuracy for recall like other models, and this gives proportional sampling the highest macro F1.

Regarding fairness, we found an increase in EOD compared with baseline and oversampling. This arises because each multicategory is represented on the same baseline counts. This can improve the model fairness because the model now has a bigger picture of categories and makes better predictions and ultimately fairer decisions. PC-SMOTE and PC-ADASYN play pivotal roles in significantly reducing bias in model predictions. This consistency confirmed the robustness of these methods across different datasets. Notably, neither method compromises accuracy while both enhance fairness, illustrating their effectiveness in handling the trade-offs typically associated with predictive modeling. These results demonstrate the strong interpolation capabilities inherent in PC-SMOTE and PC-ADASYN. These methods effectively reallocate samples within the feature space, especially moving those in underprivileged regions from negative to more positive decision boundaries. Such adjustments are crucial in mitigating biased outcomes and promoting equity in automated decision-making processes. The macro F1 in both models drops compared with the baseline because a higher number of samples is required for the model to perform better on generalization, which this dataset does not support. Specifically, the dataset has 700 samples for class 0 and 300 samples for class 1, which means the test set

only has 30 samples for class 1. This small number of samples made both models trade recall for precision in class 1. Notably, we saw a low recall for class 1 which ultimately leads to a low f1-score for class 1 and since macro F1 averages the two f1-scores and treats them equally, this affects the performance of both models in macro F1. Overall, the two models yield a fairer model with good accuracy compared with the baseline and other two sampling methods.

The COMPAS dataset's evaluation further validates our sampling methods' effectiveness. The distinct patterns that emerge align with those observed in the Adult Income and German Credit datasets, underscoring the robustness of our findings. Notably, oversampling and proportional sampling techniques have demonstrated substantial improvements in Equalized Odds Difference (EOD) and accuracy, while oversampling also notably improves in SP. This improvement is likely due to the unique composition and balance within the COMPAS dataset, unlike the other datasets in which the classes are imbalanced. The success of oversampling and proportional sampling in this context can be attributed to the balanced nature of the dataset, which allows repeated duplication of existing rows (sampling techniques employed by these methods) to enhance the dataset without introducing a significant skew towards any particular class. This method effectively augments the representation of all classes and the protected attributes in a balanced form, making these techniques particularly effective for datasets where the feature domains contribute equally to predictions and where initial class distributions do not suffer from severe imbalance. This can further be verified from their macro F1 as none of the models is trading precision for recall. The improvement of SP in oversampling can be attributed to the higher number of samples in oversampling in comparison with proportional sampling. Regarding PC-SMOTE and PC-ADASYN, these algorithms show an improvement over baseline in both accuracy and EOD. These trends follow those in the previous results. One notable thing in this results in the large drop in SP which can be attributed to our new label that was generated to make the dataset to be skewed towards the negative class. These results show the difficulty in optimizing for two or more fairness metrics at a time and how this optimization can affect each other.

6.2. Impact of Tree Depth on Fairness and Accuracy

In this study, the impact of decision tree depth on model performance was also investigated, specifically examining how variations in tree depth influence accuracy and EOD. Understanding the depth's effect is crucial as it provides insight into the effects ranging from underfitting to overfitting and helps identify the optimal complexity level at which both accuracy and fairness are maximized. Initially, the decision tree was allowed to grow without constraints to its full depth which on average was about 30 branches. The tree was then examined visually to deduce the maximum depth excluding the nonsplitting branches. To analyze the effects of tree depth systematically, the maximum depth of the trees was allowed to vary from 1 to 30. Each depth limit was evaluated using ten-fold cross-validation to ensure the robustness and generalizability of the findings.

For each configuration of tree depth, the accuracy and EOD were measured on the test set. Additionally, 95% confidence intervals were calculated for the metrics across the ten folds. This statistical analysis highlighted the depth at which the decision tree balanced the trade-off between accuracy and fairness while also considering the underlying statistical bias–variance tradeoff. By doing so, it was possible to pinpoint the “sweet spot”—a delicate point where the decision tree maintains high predictive accuracy without compromising on fairness, effectively countering the often-cited trade-off presented in previous literature. Figures 3–7 show the plots of accuracy and EOD against maximum depth for each of the five sampling methods on the Adult Income dataset.

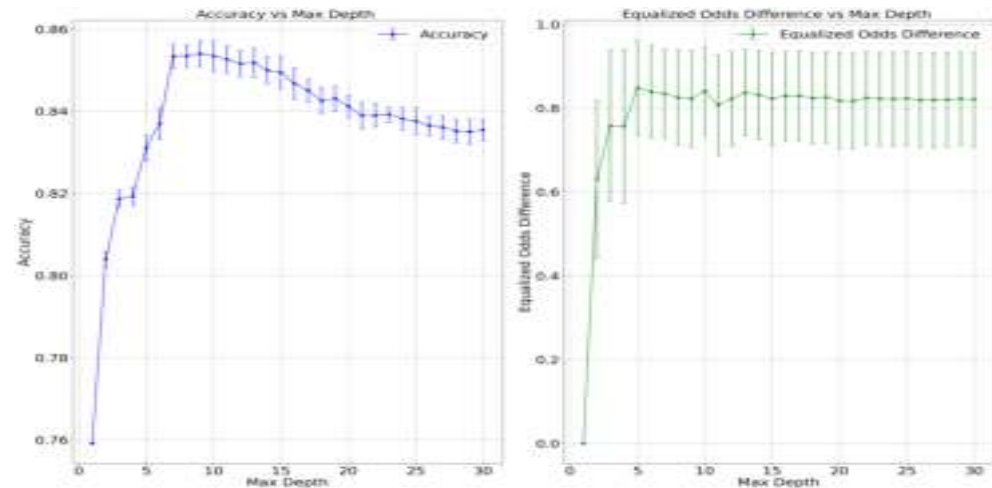


Figure 3. Plots of Adult Income using no sampling, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

Based on results such as those shown in Figures 3 and 4, there is a notable initial increase in accuracy as maximum depth increases for both the no-sampling and the over-sampling methods. However, both methods exhibit a decline in accuracy from a depth of 10 onwards, suggesting the onset of overfitting. Correspondingly, the EOD decreases sharply with increasing depth up to about depth 10, beyond which it stabilizes. This pattern indicates that while deeper trees initially improve fairness, they eventually reach a threshold beyond which no further gains are observed. Recalling that the fairness goal was to minimize EOD, a key observation is that setting the maximum depth between three and five strikes an optimal balance between achieving high accuracy and maintaining low EOD.

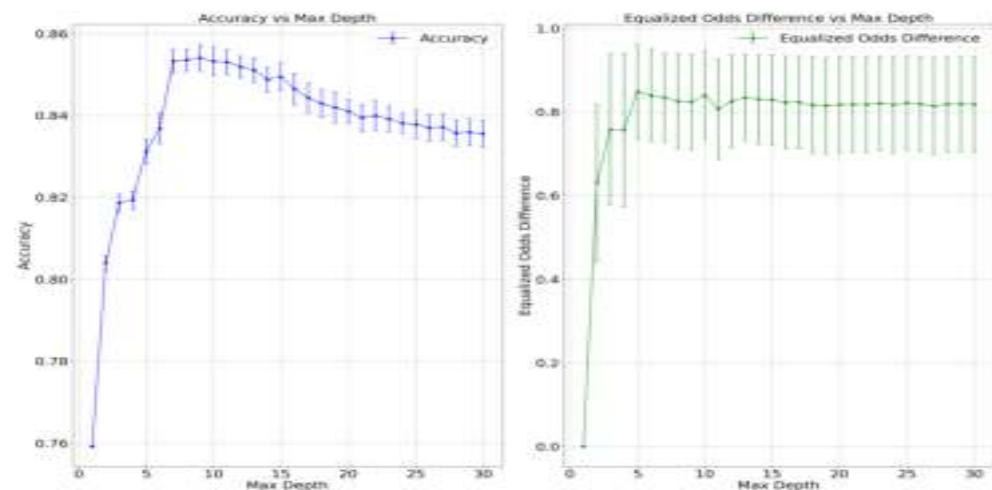


Figure 4. Plots of Adult Income using oversampling, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

When considering the results shown in Figure 5, the proportional sampling method continually increases accuracy with tree depth, peaking at a depth of about 26. Conversely, the EOD initially increases before decreasing and stabilizing at a depth of around 15. The wide confidence intervals observed in the EOD metric suggest significant variability in fairness outcomes. This finding underscores the importance of selecting a depth that minimizes variability in fairness while maximizing accuracy.

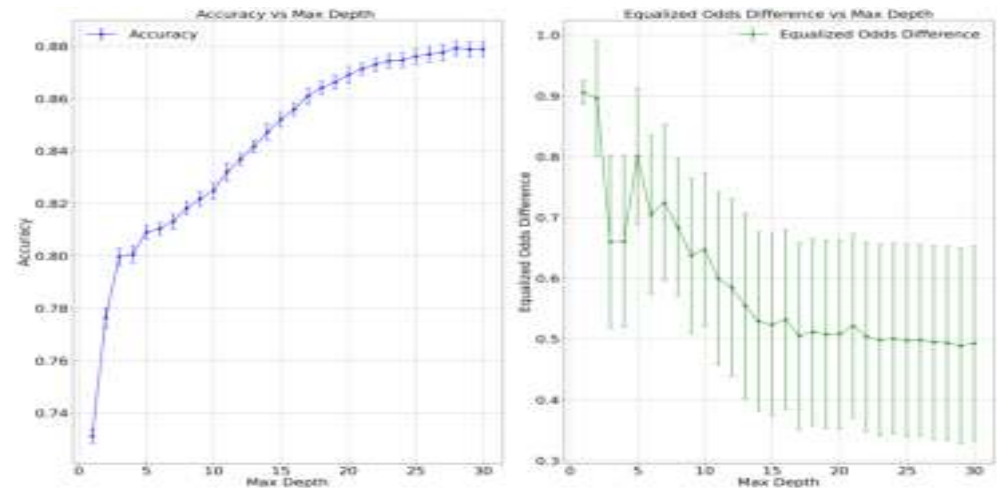


Figure 5. Plots of Adult Income using proportional sampling, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

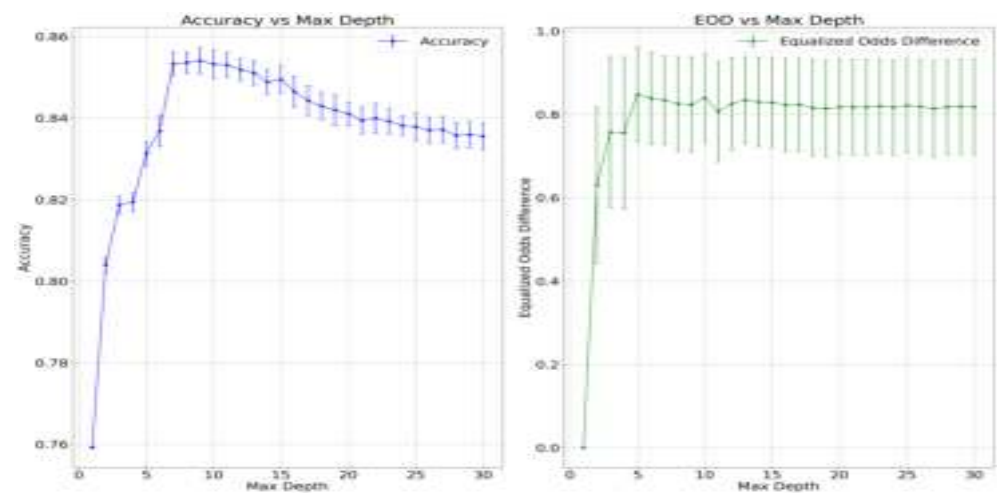


Figure 6. Plots of Adult Income using PC-SMOTE, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

Figures 6 and 7, representing the results using PC-SMOTE and PC-ADASYN, respectively, exhibit slight downward trends in accuracy, which improve briefly before descending again—a pattern indicative of overfitting at greater depths. EOD metrics for these methods show initial stability at lower depths, surge at mid-level depths, and decline, suggesting complex interactions between synthetic sample generation and decision boundary delineation. Given these observations, a maximum depth of 3 was chosen for our experiments, as it represents a “sweet spot” where both accuracy and EOD are optimized.

Given these results, one conclusion is to challenge the often presumed trade-off between accuracy and fairness by demonstrating that our PC-ADASYN method consistently outperforms baselines across all three datasets in terms of both accuracy and fairness. This finding is significant, as it suggests enhancing model fairness without sacrificing accuracy with appropriate sampling methods and model tuning is possible. However, our analysis also reveals scenarios where adjustments to model complexity, specifically the maximum depth of decision trees, can enhance accuracy at the expense of fairness, as indicated by increases in Equalized Odds Difference (EOD). It is expected, however, that coupling sampling methods with inprocessing methods such as fairness-based regularization may offset these effects. These decisions highlight researchers’ discretionary power in balancing model performance metrics depending on their study’s specific objectives and constraints. The quantity of sample and time complexity is like every other sampling method. As the

sample quantity increase, the time complexity increases but overall, the sampling methods have the same time complexity as their underlying algorithms because they have the same functionality.

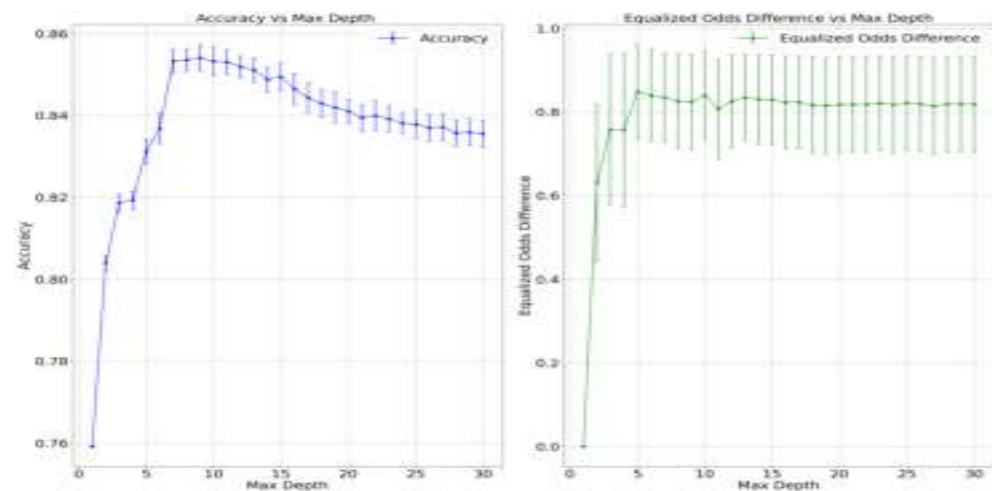


Figure 7. Plots of Adult Income using PC-ADASYN, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

Moreover, our results underscore the complexities of simultaneously optimizing multiple fairness metrics. For instance, efforts to improve Statistical Parity (SP) by favoring more positive predictions for each protected group in the COMPAS dataset led to an inadvertent reduction in negative predictions. This shift adversely impacted the False Positive Rate (FPR), a component of EOD, thereby worsening the EOD metric as SP improved. This phenomenon illustrates the inherent mathematical tensions between fairness metrics, where optimizing one can detrimentally affect another. The COMPAS dataset, with its nearly balanced class distribution, provides a concrete example of how dataset characteristics can influence the behavior of fairness metrics. Optimizing SP in this context implies a skewed measurement of fairness, particularly where inherent differences exist between groups in protected attributes. This is supported by literature indicating that SP may not adequately account for group differences, potentially leading to misleading conclusions about a model's fairness [63].

7. Limitations and Future Work

The very nature of this study is such that it is not possible to address all of the issues surrounding fairness and the so-called fairness–performance tradeoff. As such, there exist limitations in the work reported here. Even so, it is our hope and intent for the work reported here to suggest additional avenues of exploration in this important area.

One limitation of this study is that our sampling method was not specifically designed to optimize for arbitrary fairness metrics. Stated another way, since often inherent tradeoffs exist between the available set of fairness metrics, the decision was made to focus on an approach that was metric agnostic, recognizing that the results could have differed for other metrics. This is also part of the reason why we saw different behaviors between EOD and SP.

In addition, it is acknowledged that, while the underlying ML method should not be relevant to the method proposed, this has not actually been tested. Therefore, in the future, this research will be extended by considering the impacts of other ML algorithms such as logistic regression, fuzzy ID3, K-nearest neighbor, and ensemble methods such as random forests or gradient-boosted trees to assess the generalizability of our new sampling methods. The purpose of such a study would be to verify that our methods are independent of the ML algorithm employed. Furthermore, this would help validate whether the observed improvements in fairness and accuracy are model-specific or can be universally applied.

Additionally, it is acknowledged that only three distinct datasets were considered—datasets that have been studied extensively in the field. This raises a concern that methods are being tailored to these data rather than addressing the broader issue of fairness in ML. To address this, experiments with larger and more diverse datasets are planned to provide deeper insights into the scalability and robustness of our techniques. Another area for future work is to refine our multicategory sampling approach by incorporating more granular subdivisions of protected categories, potentially revealing subtler biases and providing a more nuanced understanding of fairness.

Finally, it is recognized that alternative methods have been proposed for bias mitigation, and these methods have not been studied in this work at all. Future work would entail comparisons with more sampling strategies. A more direct comparison of the proposed methods with inprocessing and postprocessing methods will be conducted. For example, incorporating inprocessing methods, such as regularization [22], or a postprocessing method, such as the Randomized Threshold Optimizer [64], will be explored as possible means to obtain further improvements in both fairness and performance.

8. Conclusions

In this study, the issue of bias in ML predictions was investigated, and a method was developed based on combining protected variables into a new multicategory. In particular, the focus was on the question that has been suggested in the literature of a bias–performance tradeoff and seeking a method to mitigate this tradeoff. The proposed new multicategory approach reflects the multifaceted identity of individuals, acknowledging the complex interplay of attributes that define real-world scenarios. Given the inherent imbalance in this multicategory, four sampling methods tailored to these complex categorizations, rather than traditional class labels, were developed. For purposes of applying a baseline classifier, decision trees were trained, and the effectiveness of these methods was evaluated using three datasets that are often employed in fairness studies. The performance of the methods was compared against baseline methods of no sampling, using accuracy, macro F1, Equalized Odds Difference (EOD), and Statistical Parity (SP) as the evaluation metrics.

The results of the experiments indicate that two of the newly developed sampling techniques—PC-SMOTE and PC-ADASYN—successfully enhance fairness without compromising accuracy. Remarkably, in some cases, these methods also improved accuracy, thus providing evidence counter to the popular claims of a fairness–performance tradeoff. Further analysis of the impact of maximum tree depth on model performance revealed that, while increasing depth initially boosts accuracy, it eventually leads to a decline. Conversely, increasing depth adversely affects fairness, highlighting the challenge of balancing complexity with equity. However, optimal tree depths were identified that simultaneously enhance accuracy and EOD, underscoring the possibility of achieving equity without sacrificing performance.

Ultimately, these findings challenge prevailing notions of an implicit performance–fairness tradeoff within bias mitigation research, suggesting that carefully designed bias mitigation strategies have the ability to sidestep this trade-off. Our approach sets a new precedent for developing more equitable predictive algorithms by redefining how protected attributes are utilized in model training.

Author Contributions: Conceptualization, G.P. and J.S.; formal analysis, G.P. and J.S.; investigation, G.P.; methodology, G.P.; software, G.P.; supervision, J.S.; validation, G.P. and J.S.; visualization, G.P.; writing—original draft, G.P.; writing—review and editing, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets and code implemented in this research work have been uploaded to <https://github.com/horlahsunbo/New-folder> (accessed on 10 June 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Maina, I.W.; Belton, T.D.; Ginzberg, S.; Singh, A.; Johnson, T.J. A decade of studying implicit racial/ethnic bias in healthcare providers using the implicit association test. *Soc. Sci. Med.* **2018**, *199*, 219–229. [\[CrossRef\]](#)
2. Salimi, B.; Rodriguez, L.; Howe, B.; Suciu, D. Causal database repair for algorithmic fairness. In Proceedings of the 2019 International Conference on Management of Data, Amsterdam, The Netherlands, 30 June–5 July 2019; pp. 793–810.
3. Kordzadeh, N.; Ghasemaghaei, M. Algorithmic Bias: Review, Synthesis, and Future Research Directions. *Eur. J. Inf. Syst.* **2022**, *31*, 388–409. [\[CrossRef\]](#)
4. Pessach, D.; Shmueli, E. A review on fairness in machine learning. *ACM Comput. Surv.* **2022**, *55*, 1–44. [\[CrossRef\]](#)
5. Aghaei, S.; Azizi, M.J.; Vayanos, P. Learning optimal and fair decision trees for non-discriminative decision-making. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 1418–1426. [\[CrossRef\]](#)
6. Kamiran, F.; Calders, T. Data preprocessing techniques for classification without discrimination. *Knowl. Inf. Syst.* **2012**, *33*, 1–33. [\[CrossRef\]](#)
7. Calmon, F.; Wei, D.; Vinzamuri, B.; Ramamurthy, K.N.; Varshney, K.R. Optimized pre-processing for discrimination prevention. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
8. Shahbazi, N.; Lin, Y.; Asudeh, A.; Jagadish, H.V. Representation bias in data: A survey on identification and resolution techniques. *ACM Comput. Surv.* **2023**, *55*, 1–39. [\[CrossRef\]](#)
9. Chen, Z.; Zhang, J.M.; Sarro, F.; Harman, M. A comprehensive empirical study of bias mitigation methods for machine learning classifiers. *ACM Trans. Softw. Eng. Methodol.* **2023**, *32*, 1–30. [\[CrossRef\]](#)
10. Perzynski, A.; Berg, K.A.; Thomas, C.; Cembali, A.; Smith, T.; Shick, S.; Gunzler, D.; Sehgal, A.R. Racial discrimination and economic factors in redlining of Ohio neighborhoods. *Bois Rev. Soc. Sci. Res. Race* **2023**, *20*, 293–309. [\[CrossRef\]](#)
11. Steil, J.P.; Albright, L.; Rugh, J.S.; Massey, D.S. The social structure of mortgage discrimination. *Hous. Stud.* **2018**, *33*, 759–776. [\[CrossRef\]](#)
12. Salgado, J.F.; Moscoso, S.; García-Izquierdo, A.L.; Anderson, N.R. *Shaping Inclusive Workplaces through Social Dialogue*; Springer: Berlin/Heidelberg, Germany, 2017.
13. Leavy, S.; Meaney, G.; Wade, K.; Greene, D. Mitigating gender bias in machine learning data sets. In Proceedings of the Bias and Social Aspects in Search and Recommendation: First International Workshop, BIAS 2020, Lisbon, Portugal, 14 April 2020; pp. 12–26.
14. Hort, M.; Chen, Z.; Zhang, J.M.; Harman, M.; Sarro, F. Bias mitigation for machine learning classifiers: A comprehensive survey. *ACM J. Responsible Comput.* **2023**, *1*, 1–52. [\[CrossRef\]](#)
15. Fahse, T.; Huber, V.; van Giffen, B. Managing bias in machine learning projects. In *Innovation Through Information Systems: Volume II: A Collection of Latest Research on Technology Issues*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 94–109.
16. Zhang, Z.; Neill, D.B. Identifying Significant Predictive Bias in Classifiers. *arXiv* **2017**, arXiv:1611.08292. [\[CrossRef\]](#)
17. Zheng, Z.; Cai, Y.; Li, Y. Oversampling method for imbalanced classification. *Comput. Inform.* **2015**, *34*, 1017–1037.
18. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [\[CrossRef\]](#)
19. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; pp. 1322–1328.
20. Janssen, P.; Sadowski, B.M. Bias in Algorithms: On the trade-off between accuracy and fairness. In Proceedings of the 23rd Biennial Conference of the International Telecommunications Society, Gothenburg, Sweden, 21–23 June 2021.
21. Hardt, M.; Price, E.; Srebro, N. Equality of opportunity in supervised learning. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.
22. Kamishima, T.; Akaho, S.; Asoh, H.; Sakuma, J. Fairness-aware classifier with prejudice remover regularizer. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, 24–28 September 2012; pp. 35–50.
23. Calders, T.; Žliobaitė, I. Why unbiased computational processes can lead to discriminative decision procedures. In *Discrimination and Privacy in the Information Society: Data Mining and Profiling in Large Databases*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 23–33.
24. Burt, A. How to Fight Discrimination in AI. Harvard Business Review. 2020 Available online: <https://hbr.org/2020/08/how-to-fight-discrimination-in-ai> (accessed on 12 July 2024).
25. Siegler, A.; Admussen, W. Discovering Racial Discrimination by the Police. *Northwestern Univ. Law Rev.* **2021**, *115*, 987–1054.
26. Grabowicz, P.; Perello, N.; Mishra, A. How to Train Models that Do Not Propagate Discrimination? Equate and Machine Learning Blog, University of Massachusetts, Amherst. 2022. Available online: <https://groups.cs.umass.edu/equate-ml/2022/04/07/how-to-train-models-that-do-not-propagate-discrimination/> (accessed on 12 July 2024).
27. Khani, F.; Liang, P. From Discrimination in Machine Learning to Discrimination in Law, Part 1: Disparate Treatment. Stanford AI Lab Blog. 2022. Available online: https://ai.stanford.edu/blog/discrimination_in_ML_and_law/ (accessed on 12 July 2024).
28. Shahriari, K.; Shahriari, M. IEEE standard review—Ethically aligned design: A vision for prioritizing human wellbeing with artificial intelligence and autonomous systems. In Proceedings of the IEEE Canada International Humanitarian Technology Conference (IHTC), Toronto, ON, Canada, 21–22 July 2017; pp. 197–201.

29. European Commission; Directorate-General for Communications Networks, Content and Technology. Ethics Guidelines for Trustworthy AI. Publications Office. 2019. Available online: <https://op.europa.eu/en/publication-detail/-/publication/d3988569-0434-11ea-8c1f-01aa75ed71a1/language-en/format-PDF/source-337437547> (accessed on 10 June 2024)
30. Ebers, M.; Hoch, V.R.S.; Rosenkranz, F.; Ruschemeier, H.; Steinrötter, B. The European Commission's Proposal for an Artificial Intelligence Act—A Critical Assessment by Members of the Robotics and AI Law Society (RAILS). *J* **2021**, *4*, 589–603. [\[CrossRef\]](#)
31. Hajian, S.; Domingo-Ferrer, J. A methodology for direct and indirect discrimination prevention in data mining. *IEEE Trans. Knowl. Data Eng.* **2012**, *25*, 1445–1459. [\[CrossRef\]](#)
32. Fish, B.; Kun, J.; Lelkes, Á.D. A confidence-based approach for balancing fairness and accuracy. In Proceedings of the 2016 SIAM International Conference on Data Mining, Miami, FL, USA, 5–7 May 2016; pp. 144–152.
33. Hajian, S. Simultaneous Discrimination Prevention and Privacy Protection in Data Publishing and Mining. *arXiv* **2013**, arXiv:1306.6805. [\[CrossRef\]](#)
34. Sondeck, L.P.; Laurent, M.; Frey, V. The Semantic Discrimination Rate Metric for Privacy Measurements which Questions the Benefit of t -closeness over l -diversity. In Proceedings of the 14th International Conference on Security and Cryptography, Madrid, Spain, 24–26 July 2017; Volume 6, pp. 285–294.
35. Ruggieri, S. Using t -closeness anonymity to control for non-discrimination. *Trans. Data Priv.* **2014**, *2*, 99–129.
36. Romano, Y.; Bates, S.; Candes, E. Achieving equalized odds by resampling sensitive attributes. In Proceedings of the Advances in Neural Information Processing Systems, Online, 6–12 December 2020; pp. 361–371.
37. Peng, K.; Chakraborty, J.; Menzies, T. Fairmask: Better fairness via model-based rebalancing of protected attributes. *IEEE Trans. Softw. Eng.* **2022**, *49*, 2426–2439. [\[CrossRef\]](#)
38. Dhar, P.; Gleason, J.; Roy, A.; Castillo, C.D.; Chellappa, R. PASS: Protected attribute suppression system for mitigating bias in face recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 15087–15096.
39. Krasnakis, E.; Spyromitros-Xioufis, E.; Papadopoulos, S.; Kompatsiaris, Y. Adaptive sensitive reweighting to mitigate bias in fairness-aware classification. In Proceedings of the World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 853–862.
40. Rančić, S.; Radovanović, S.; Delibašić, B. Investigating oversampling techniques for fair machine learning models. In Proceedings of the Decision Support Systems XI: Decision Support Systems, Analytics and Technologies in Response to Global Crisis Management: 7th International Conference on Decision Support System Technology, ICDSSST 2021, Loughborough, UK, 26–28 May 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 110–123.
41. Yan, S.; te Kao, H.; Ferrara, E. Fair class balancing: Enhancing model fairness without observing sensitive attributes. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, Ireland, 19–23 October 2010; pp. 1715–1724.
42. HuZhang, B.; Lemoine, B.; Mitchell, M. Mitigating unwanted biases with adversarial learning. In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, New Orleans, LA, USA, 2–3 February 2018; pp. 335–340.
43. Celis, L.E.; Huang, L.; Keswani, V.; Vishno, N.K. Classification with fairness constraints: A meta-algorithm with provable guarantees. In Proceedings of the Conference on Fairness, Accountability, and Transparency, Atlanta, GA, USA, 29–31 January 2019; pp. 319–328.
44. Zafar, M.B.; Valera, I.; Rodriguez, M.G.; Gummadi, K.P. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 1171–1180.
45. Agarwal, A.; Beygelzimer, A.; Dudík, M.; Langford, J.; Wallach, H. A reductions approach to fair classification. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 60–69.
46. Lowy, A.; Baharlouei, S.; Pavan, R.; Razaviyayn, M.; Beirami, A. A Stochastic Optimization Framework for Fair Risk Minimization. *Trans. Mach. Learn. Res.* **2022**. [\[CrossRef\]](#)
47. Spinelli, I.; Scardapane, S.; Hussain, A.; Uncini, A. Fairdrop: Biased edge dropout for enhancing fairness in graph representation learning. *IEEE Trans. Artif. Intell.* **2021**, *3*, 344–354. [\[CrossRef\]](#)
48. Hort, M.; Zhang, J.M.; Sarro, F.; Harman, M. Fairea: A model behaviour mutation approach to benchmarking bias mitigation methods. In Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Athens, Greece, 23–28 August 2012; pp. 994–1006.
49. Bhaskaruni, D.; Hu, H.; Lan, C. Improving Prediction Fairness via Model Ensemble. In Proceedings of the IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, OR, USA, 4–6 November 2019; pp. 1810–1814.
50. Iosifidis, V.; Fetahu, B.; Ntoutsi, E. Fae: A fairness-aware ensemble framework. In Proceedings of the IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 1375–1380.
51. Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; Zemel, R. Fairness through awareness. In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, Cambridge, MA, USA, 8–10 January 2012; pp. 214–226.
52. Pleiss, G.; Raghavan, M.; Wu, F.; Kleinberg, J.; Weinberger, K.Q. On fairness and calibration. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–8 December 2017.
53. Karimi-Haghighi, M.; Castillo, C. Enhancing a recidivism prediction tool with machine learning: Effectiveness and algorithmic fairness. In Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, São Paulo, Brazil, 21–25 June 2017; pp. 210–214.

54. Friedler, S.A.; Scheidegger, C.; Venkatasubramanian, S. The (im)possibility of fairness: Different value systems require different mechanisms for fair decision making. *Commun. ACM* **2021**, *64*, 136–143. [CrossRef]
55. García, V.; Sánchez, J.S.; Mollineda, R.A. On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowl.-Based Syst.* **2012**, *25*, 13–21. [CrossRef]
56. Breiman, L. *Classification and Regression Trees*; Routledge: London, UK, 2017.
57. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
58. Kelly, M.; Longjohn, R.; Nottingham, K. The UCI Machine Learning Repository. 2024. Available online: <https://archive.ics.uci.edu> (accessed on 10 June 2024).
59. Becker, B.; Kohavi, R. Adult. UCI Machine Learning Repository. 1996. Available online: <https://archive.ics.uci.edu/dataset/2/adult> (accessed on 10 June 2024). [CrossRef]
60. Hofmann, H.; Statlog (German Credit Data). UCI Machine Learning Repository. 1994. Available online: <https://archive.ics.uci.edu/dataset/144/statlog+german+credit+data> (accessed on 10 June 2024). [CrossRef]
61. Dressel, J.; Farid, H. The accuracy, fairness, and limits of predicting recidivism. *Sci. Adv.* **2018**, *4*, eaao5580. [CrossRef]
62. Huang, Q.; Mao, J.; Liu, Y. An improved grid search algorithm of SVR parameters optimization. In Proceedings of the 2012 IEEE 14th International Conference on Communication Technology, Chengdu, China, 9–11 November 2012; pp. 1022–1026.
63. Caton, S.; Haas, C. Fairness in machine learning: A survey. *ACM Comput. Surv.* **2023**, *56*, 1–38. [CrossRef]
64. Alabdulmohsin, I.; Lucic, M. A Near-Optimal Algorithm for Debiasing Trained Machine Learning Models. In Proceedings of the 35th Conference on Neural Information Processing Systems, Online, 6–14 December 2021.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Secure Processing and Distribution of Data Managed on Private InterPlanetary File System Using Zero-Knowledge Proofs

Kyohei Shibano ^{1,*} , Kensuke Ito ¹, Changhee Han ², Tsz Tat Chu ², Wataru Ozaki ² and Gento Mogi ¹¹ Department of Technology Management for Innovation, School of Engineering, The University of Tokyo, Tokyo 113-8656, Japan² Callisto Inc., Tokyo 171-0022, Japan

* Correspondence: shibano@tmi.t.u-tokyo.ac.jp

Abstract: In this study, a new data-sharing method is proposed that uses a private InterPlanetary File System—a decentralized storage system operated within a closed network—to distribute data to external entities while making its authenticity verifiable. Among the two operational modes of IPFS, public and private, this study focuses on the method for using private IPFS. Private IPFS is not open to the general public; although it poses a risk of data tampering when distributing data to external parties, the proposed method ensures the authenticity of the received data. In particular, this method applies a type of zero-knowledge proof, namely, the Groth16 protocol of zk-SNARKs, to ensure that the data corresponds to the content identifier in a private IPFS. Moreover, the recipient's name is embedded into the distributed data to prevent unauthorized secondary distribution. Experiments confirmed the effectiveness of the proposed method for an image data size of up to 120 × 120 pixels. In future studies, the proposed method will be applied to larger and more diverse data types.

Keywords: IPFS; zero-knowledge proof; circom; zk-SNARKs; private IPFS; data distribution; data processing; data security



Citation: Shibano, K.; Ito, K.; Han, C.; Chu, T.T.; Ozaki, W.; Mogi, G. Secure Processing and Distribution of Data Managed on Private InterPlanetary File System Using Zero-Knowledge Proofs. *Electronics* **2024**, *13*, 3025. <https://doi.org/10.3390/electronics13153025>

Academic Editor: Aryya Gangopadhyay

Received: 13 June 2024

Revised: 14 July 2024

Accepted: 20 July 2024

Published: 31 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Decentralized systems are robust because they lack a single point of failure; therefore, they are widely applied across enterprise sectors including cryptocurrency, supply chain management, financial services, and digital identity. To store large-sized data such as images, these systems require storage functions that are inherently decentralized. Blockchain, commonly used in conjunction, typically handles smaller data sizes such as transaction histories and operates as a ledger database. The InterPlanetary File System (IPFS) is a prominent decentralized storage system that stores data across multiple nodes to enhance data availability. The IPFS has two variants: public IPFS, wherein the data can be stored by any user with unrestricted access, and private IPFS, wherein a closed network accessible only within specific organizations or groups is established, offering enhanced privacy and security.

When storing data in IPFS, understanding the differences between public IPFS and private IPFS is crucial. Public IPFS allows anyone to access data, while private IPFS is accessible only within specific organizations or groups, enhancing privacy and security. When storing sensitive information, such as confidential data, in public IPFS, applying an appropriate encryption scheme is vital to ensure data protection. By contrast, private IPFS provides higher security for data storage, because it is accessible only within a closed network.

Particularly for organizations such as corporations or healthcare institutions, storing data in public IPFS, despite using strong encryption technologies, carries inherent risks. Moreover, the potential for data leaks due to operational errors exists persistently in such cases; although data is encrypted, it is exposed to the world, rendering it vulnerable to brute force attacks and other security threats.

Regarding accessibility, public IPFS allows general users to directly access and retrieve data. However, in private IPFS, data must be received from members of the organization or group constituting the network. During this process, if data is tampered, then users may be unable to detect it. Therefore, trusting the intermediaries responsible for handling data transfer in such cases becomes mandatory. To address the aforementioned trust issue, a new method is proposed herein for distributing data stored in a private IPFS to external entities while making its authenticity verifiable. The Groth16 [1] protocol of zk-SNARKs, a type of ZKP, is applied to data stored in a private IPFS to ensure the authenticity of the data. Moreover, the recipient's information is embedded into the distributed data to prevent unauthorized secondary distribution. The proposed method of data sharing is important because it is tailored to the private IPFS case.

The differences in several aspects, including security and accessibility, when storing data in public IPFS and private IPFS within the enterprise domain are summarized in Table 1. This study proposes solutions to the threats associated with private IPFS.

Table 1. Comparison between public and private IPFS in the enterprise domain.

	Public IPFS	Private IPFS
Trust Model	Trustless	Requires trust in the operating group
Access Restrictions	Accessible by anyone	Accessible only within the operating group
Data Leakage Risk	Constant risk of leakage due to user error	Low risk of leakage within a closed network
Handling of Confidential Information	Requires proper encryption	Data stored in IPFS does not require high-level encryption itself; there is a trust point when passing data to users
Brute Force Attack Risk	Always present	Low
Data Retrieval Method	Direct access by users	Data received from members of the organization or group
Threats	Requires encryption that prevents decryption by unauthorized users	There is a risk of tampering when transferring data to users

The remainder of this paper is organized as follows. Section 2 presents related prior research. Section 3 outlines the fundamental technologies, i.e., ZKP and zk-SNARKs. Section 4 describes the structure of the proposed method, while Section 5 outlines the potential applications of this method. Section 6 presents the implementation of this method, while Section 7 discusses the experiments performed to verify the effectiveness of the implementation. Section 8 presents a discussion of the experimental results, while Section 9 presents the conclusions of the paper and an outline of future challenges.

2. Related Studies

Existing decentralized systems use IPFS, particularly in combination with blockchain technology. Kumar et al. [2] proposed a method for securely managing medical data by integrating IPFS with a blockchain. Azbeg et al. [3] specifically suggested a system that managed and stored medical data using private IPFS and a permissioned blockchain by employing proxy re-encryption to ensure secure decryption by designated doctors. When a physician receives some patient's data, he/she obtains the re-encrypted data via a hospital. Hossan et al. [4] also proposed a system to securely record information for ride-sharing services using IPFS and a private blockchain.

Focusing on controlling the distribution of data managed by IPFS, Lin et al. [5] proposed a system for protecting private data using improved IPFS combined with a blockchain. This system recorded file metadata and accessed permissions on the blockchain, enabling users to control file sharing. Moreover, the system implemented efficient management features using smart contracts, thereby enhancing data security and management

flexibility. Battah et al. [6] developed a system that used multiparty authentication (MPA), proxy re-encryption, and smart contracts on a blockchain for decentralized access control of encrypted data stored in IPFS. Huang et al. [7] introduced a trusted IPFS proxy to realize access control and group key management for encrypted data stored in IPFS. Sun et al. [8] proposed a system that allowed only individuals with appropriate attributes to decrypt encrypted data stored in IPFS using a ciphertext policy attribute-based encryption system, facilitating efficient medical information management. Kang et al. [9] enabled the distribution of data managed using private IPFS and a private blockchain to external users using named data network (NDN). Furthermore, Uddin et al. [10] proposed a file-sharing system that used IPFS and public key infrastructure (PKI) technology without requiring a trusted third party.

Several studies have used ZKP for data distribution. For instance, Li et al. [11] proposed a privacy-preserving traffic management system that combined noninteractive zero-knowledge range proofs with a blockchain. A prototype using Hyperledger Fabric and Hyperledger Ursa met the data privacy requirements for real-time traffic management.

This study proposes a method for appropriately processing and distributing data managed within private IPFS to users outside the network, thereby offering a different approach than those proposed in previous studies. Some studies have adopted proxy re-encryption as an appropriate method for data storage and distribution in IPFS [3,6]. Using this method, distributed data can be re-encrypted to be decrypted with the recipient's private key. Moreover, when storing data in IPFS, recording the hash value of the pre-encrypted data on the blockchain allows recipients to verify the correctness of their received data after decryption. However, this method cannot handle cases where data is processed, such as embedding the recipient's name into the decrypted data, as in this study.

3. Zero-Knowledge Proof

ZKP is a cryptographic protocol that allows a prover to prove the validity of a proposition to a verifier without disclosing any additional information other than the validity of the proposition. The proposition of this study is that the data provided to an external entity is generated based on a given CID. Our goal is to allow a member of private IPFS (prover) to prove this proposition to an external entity (verifier as the recipient of the data) without disclosing any other important information (such as IPFS access rights and encryption keys).

ZKP, specifically the Groth16 protocol of zk-SNARKs used in this study, begins with a trusted setup where both parties establish public parameters that are crucial for the secure generation and verification of proofs. In the ZKP scheme, we first generate a circuit that describes the process for which a proof is intended. The circuit includes the conditions to be verified such as the existence of a CID. Through the ZKP scheme, cryptographic keys—specifically a proving key and a verification key—are created. These keys are crucial for creating a proof for the circuit and its verification. Using the proving key and input data, the prover generates a proof that reveals the validity of the output data against the conditions specified in the circuit. The verifier then uses the verification key to check the proof and the output data. If the proof is valid, this confirms the integrity of data without exposing any underlying information.

In this study, Groth16 processing is performed using circom [12,13] and snarkjs [14]. The process flow is summarized in Figure 1.

zk-SNARKs is employed owing to its noninteractive nature and efficiency, which are particularly advantageous for systems employing smart contracts owing to low computational costs for verifying the proof. Furthermore, zk-SNARKs is known for its high computational requirements and the need for advanced PC specifications. For instance, in one of the representative applications of zk-SNARKs, Zcash [15], proof generation process takes over half a minute for a single anonymous transaction [16].

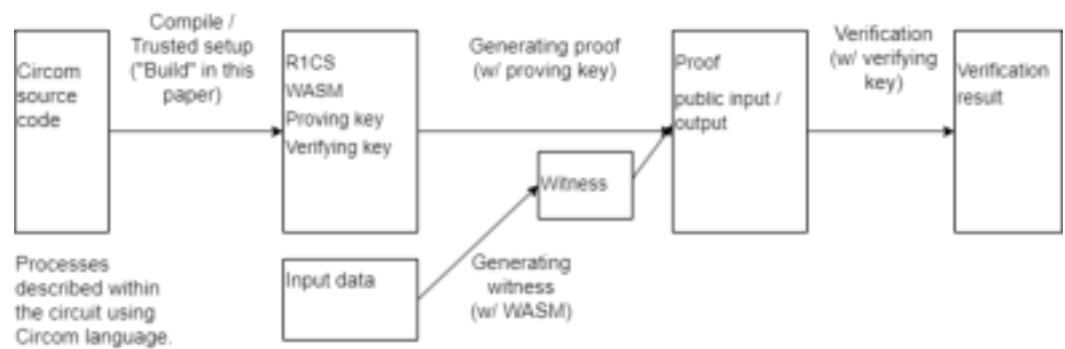


Figure 1. Zero-knowledge proof using circom.

4. Proposed System

Figure 2 presents an overview of the proposed system.

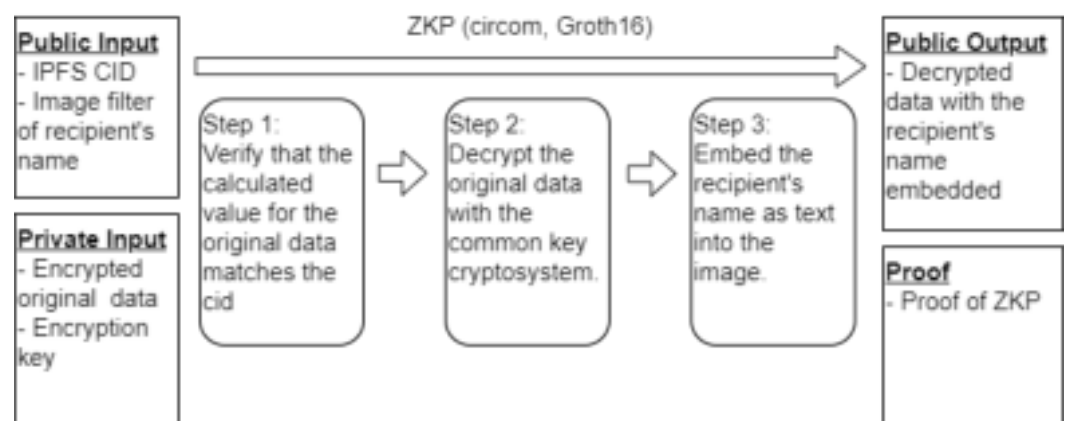


Figure 2. Process flow of ZKP: an example of image data processing.

Herein, we make the following assumptions:

- Private IPFS is operated by a limited number of members.
- Data are stored in IPFS in an encrypted format using symmetric-key cryptography.
- The encryption key is exclusively held by an individual among the members, i.e., an administrator.

The proposed system facilitates the creation of a ZKP proof through the circuit by the encryption key-holding member (equivalent to an administrator). The inputs and outputs (other than proofs) of this process are as follows:

- **Public Input:**
 - CID of the original encrypted data;
 - A filter for embedding the recipient's name in the data;
- **Private Input:**
 - Encrypted original data;
 - Encryption key;
- **Output:**
 - Decrypted data with the recipient's name embedded.

Note that the ZK-optimized implementation (Section 6.1) adds more information to the public input. In particular, the internal process of ZKP involves the following steps:

1. Calculating the CID of the original encrypted data to verify a match with the entered CID.

2. Decrypting the original data using a symmetric-key cryptography.
3. Embedding the recipient's name into the decrypted image raw data based on the filter provided in the public input.

In the proposed system, we do not solely focus on image data but use them as example data to verify the applicability of the proposed scheme. Filters are used to improve the efficiency of processing inside the circuit. As embedding name data inside the circuit is computationally intensive, a considerable portion of the image processing is performed outside the circuit in advance and a filter is created. Using public input and proof, the recipient can verify that the decrypted data (i.e., output) with their name embedded are generated from the original data (contained in the private input) managed with the CID. The recipient can check with at least one member of the network to confirm the existence of the CID in private IPFS.

In summary, the aforementioned process enables the recipient to verify the received data by performing the following tasks:

- verify that the received data were generated from the data managed with the CID of private IPFS,
- confirm using the proof that the entire process was correctly conducted without directly knowing the encrypted data or the encryption key, and
- verify that the CID exists specifically within the private IPFS by asking at least one network member.

The novelty of the proposed system is that it allows data authenticity verification by trusting at least one member of the network even if the recipient do not control the encryption key. (It is natural for the recipient to trust at least one member of a particular multimember system. If none of the members can be trusted, then there will be a marginal incentive to receive data managed by that network).

5. Potential Applications

In the medical industry, patient diagnosis data are managed across multiple medical institutions. Using the proposed system, patients can verify whether the data they receive are indeed managed in private IPFS to ensure the authenticity. An all-in-one platform is also proposed herein for the research and development of machine learning with medical images [17]. On this platform, anonymized medical images are managed in private IPFS operated by a group of medical institutions. The system allows machine learning researchers, who are external to the network, to verify whether the image data are indeed managed in the private IPFS. Moreover, by embedding the information about machine learning researchers in the image data, medical institutions can mitigate the risk of secondary distribution.

If the application is not limited to the embedding of recipient's name, the potential applications of the proposed system can be further expanded. For instance, consider a scenario where a specific company establishes private IPFS for sharing confidential documents among its group companies. If employee data are included, then concealing private data and distributing them to external entities allows these entities to confirm the association of employees with the company while ensuring that their privacy is protected. Furthermore, suppose a university has set up private IPFS to allow only academic staff access to student performance data. In this case, students can verify that their performance data received are genuinely managed in the private IPFS.

Thus, the proposed system supports a hybrid case—distributing internal data to specific external entities as necessary—prevalent in real-world settings.

6. Implementation

The proposed system was implemented to process image data using circom, a renowned tool specialized for constructing zk-SNARKs circuits. Circom enables the description of computational processes within a circuit using its unique language, and the

executable file generated after compilation can be invoked via the JavaScript library, snarkjs. This arrangement allows describing circuit processes in circom, and external processing and circuit correctness testing are performed using JavaScript. For the zk-SNARKs scheme, Groth16 was used; it is known for its relatively faster execution speed than other zk-SNARKs scheme.

In particular, we worked on two types of implementations for image data: a standard implementation using general cryptographic techniques and a ZK-optimized implementation using ZK-friendly cryptographic techniques to reduce the computation time of the circuit. These implementations were used for comparing the required computation times. ZKP circuits require considerably large computation time, even for calculations that can be easily handled by computer software (this is particularly noticeable when dealing with image data). Therefore, computational efficiency is crucial for practical use.

6.1. Standard Implementation

Section 4 describes the data input into the circuit. For simplifying the in-circuit processing, the original encrypted data were formatted as bitmap image data compliant with OS/2 standards. The first 54 bytes of the image data store information such as the width, height, and color depth of images [18]. The color depth is 8 bits and each color component in RGB is allocated one byte, resulting in a representation of 3 bytes per pixel.

Initially, the system checks whether the encrypted data, entered as a private input, matches the CID provided as a public input. If they do not match, the system signifies an error and the image data outputted as the output is a byte sequence where all values are 0x00. CID serves as crucial mechanism for uniquely identifying files and efficiently retrieving data from IPFS. CID has two versions: V0 and V1 [19]. Herein, the more flexible version CID V1 was used. CID includes a hash of the respective data, ensuring different data will have different CIDs. Typically, CID V1 is calculated using the SHA256 hash function, and the standard implementation uses SHA256 to compute CID.

The data structure of CID V1 is as shown in Table 2.

Table 2. CID V1 data structure.

Byte Position	Description	Value in Implementation
First byte	CID version	0x01
Second byte	multibase prefix	0x55: raw data
Third byte	Hash function identifier	0x12: SHA-256
Fourth byte	Hash length	0x20: 32 bytes
From fifth byte	Hash value	SHA-256 hash value (32 bytes)

The encoding for CID is conducted using Base32. Base32 encodes a sequence of bytes constructed based on this structure to generate CID.

Inside the circuit, the entered CID value is decoded from Base32 and the system checks whether the extracted hash value matches the SHA256 hash computed from the encrypted data.

Subsequently, the encrypted data are decrypted. AES-CTR is used as the encryption algorithm, which is a type of symmetric-key cryptography. The AES-CTR encryption and decryption in circom-chacha20 [20] was used. For decrypting AES-CTR encryption, the encryption key and nonce used during encryption are required. They are input into the circuit as a 256-bit key and a 128-bit nonce, respectively, as private inputs. Moreover, AES-CTR handles data volumes in multiples of 16. Therefore, if the length of the image data before encryption is not a multiple of 16, zeros (0x00) are added to the end of the data to align it with this requirement.

Finally, a filter is applied to the decrypted data to embed the recipient's name. Implementing text embedding directly within the circuit can substantially increase the computation load; therefore, a filter is created outside the circuit that performs a considerable

portion of the image processing in advance. The font used for the text representing the recipient's name is the Misaki font [21]. The filter is then used to streamline processing inside the circuit. The filter is a list of numbers where values from 0 to 255 are used to change the color of each pixel in case it differs from that of the pixels in the original image; moreover, a value of 300 indicates the color should remain as in the original image. This filter represents the position on the image where the recipient's name should be inserted. Inside the circuit, the specified pixel colors in the decrypted BMP data are changed based on this filter.

6.2. ZK-Optimized Implementation

ZK-optimized implementation changes the hash function, encryption technology, and in-circuit processing to the standard ZK-friendly encryption implementation. This implementation enhances the computational efficiency and does not evaluate the difference in computation speeds between ZK-friendly encryption and general encryption. Therefore, in-circuit processing was also modified.

Poseidon hash [22] was used as the hash function for computing CID. Notably, using the Poseidon hash for CIDs is not officially supported; therefore, it was developed specifically for this study. Although SHA256 is commonly used in general computations, it demands considerable computation time within ZKP circuits. The Poseidon hash is implemented in circom and JavaScript (circomlib [23] and circomlibjs [24], respectively). It is computed over a finite field with a prime order and can accept up to 16 input variables. The used order is less than the maximum of 32 bytes but greater than the maximum of 31 bytes. This indicates that each of the 16 inputs must contain data not exceeding this order. In this implementation, the data targeted for hash computation are divided into 31-byte segments as input values. If the division exceeds 16 segments, the Poseidon hash is calculated for the first 16 segments. This result is added to the next 15 segments of data for a subsequent Poseidon hash input. The process is repeated until all the input data are used for hash computation. Computationally, if the final input does not complete 16 segments, the missing inputs are set to zero to ensure that the computation always involves 16 inputs.

When generating CID from the Poseidon hash value, the byte sequence should follow the CID V1 data structure and be Base32-encoded. However, to further reduce computation time, this implementation omits the Base32 encoding and directly uses the Poseidon hash value as a substitute for CID. Dividing the input data into 16 segments within the circuit is computationally intensive; therefore, this division is performed outside the circuit and given as an input. In this case, the encrypted data byte sequence and the list of values for calculating the Poseidon hash are provided as public inputs, allowing the verification that both datasets represent the same information. Recipients can confirm that the data being computed for the Poseidon hash and the data being decrypted in the circuit are identical by mutually converting and checking these two values. In this case, as users can obtain the decrypted data, a concern exists regarding password leakage through brute force attacks or other means.

For encryption technology, we adopted Poseidon encryption [25] instead of AES-CTR encryption. Poseidon encryption, implemented in circom and TypeScript (poseidon-encryption-circom2 [26]), involves receiving the public key of the recipient, generating a common key, and ensuring secure encryption and decryption by both parties. In this case, however, a common key is directly generated and used for encryption and decryption. The circuit is provided with two values representing the coordinates of an elliptical curve and a nonce value as private inputs for encryption. Moreover, the filter is implemented in the same manner as in the standard implementation.

7. Evaluation

We created a sample program based on the aforementioned implementations that uses circom to describe the circuit and uses snarkjs for executing the circuit and verifying

proofs. As cryptographic libraries, circom-chacha20 [20], circomlib [23], circomlibjs [24], and poseidon-encryption-circom2 [26] were used.

The standard and ZK-optimized implementations were implemented for each circuit, and their computation times were compared during execution. White bitmap images were the target images, and the experiments were conducted using the letter “A” as the embedded character. As embedding any number of characters does not alter the processing by the filter, embedding a single character allowed for comparing the computation times. Furthermore, we varied the image sizes to measure the execution times for each circuit. The sizes used were 10×10 , 15×15 , 30×15 , 30×30 , 60×30 , 60×60 , 120×60 , 120×120 , and 180×120 pixels. The execution environment was Windows 11 with a Ryzen 9 3950X CPU and 128 GB RAM operating under Ubuntu 22.04 in a WSL2 environment.

Figure 3 shows an example image generated by the circuit, specifically for the 60×30 pixel size using the ZK-optimized implementation. The results for each image size are presented in Table 3, where nonlinear constraints indicate the number of nonlinear constraints in the circuit, build time is the time required to compile circom and output the circuit, and proof gen time is the time required to generate proofs using the circuit. As standard implementation uses AES-CTR encryption, data with 0x00 are appended at the end to ensure that the input size is a multiple of 16.

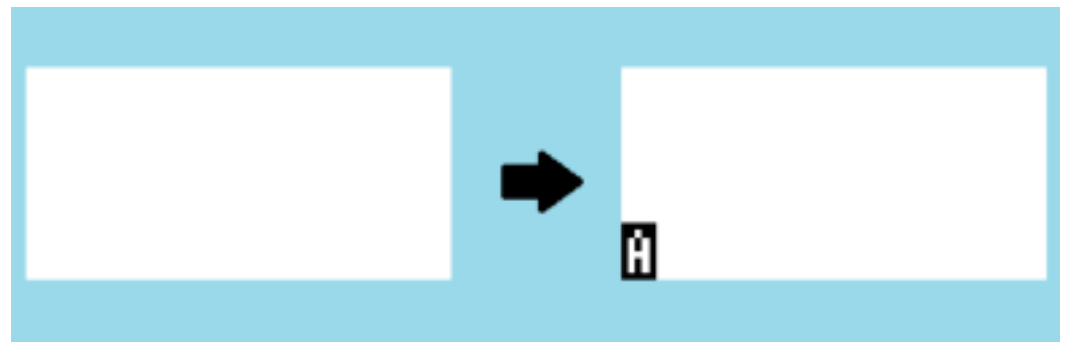


Figure 3. An image generated by the circuit for a 60×30 image size by ZK-optimized implementation.

Table 3. Comparison of the execution time of the circuit.

	Pixel	Image Size [Byte]	Nonlinear Constraints	Build Time [ms]	Proof Gen Time [ms]
Standard	10×10	384	558,341	668,220	14,377
	15×15	784	1,095,292	1,316,666	25,545
	30×15	1440	1,980,688	1,696,370	35,161
	30×30	2816	3,867,989	3,657,578	65,785
	60×30	5456	7,453,300	7,864,583	126,262
ZK-optimized	10×10	376	35,407	128,979	3076
	15×15	775	72,725	173,210	4032
	30×15	1435	134,663	275,630	5900
	30×30	2815	263,450	470,872	9733
	60×30	5455	509,375	850,612	17,571
	60×60	10,855	1,013,483	1,257,418	29,044
	120×120	43,255	4,036,913	6,754,928	96,580

In standard and ZK-optimized implementations for 60×60 pixel and 180×120 pixel image sizes, the system ran out of memory and the computation could not be completed. In ZK-optimized implementation, the number of nonlinear constraints was reduced to approximately one-tenth that of the standard implementation for the same image size.

This reduced the build and proof generation times. However, the maximum manageable image size was still only up to 120×120 pixels, which is considerably small for practical applications.

8. Discussion

Although ZK-friendly cryptographic technologies were used and in-circuit processes were optimized during ZK-optimized implementation, the maximum manageable image size was approximately 120×120 pixels. This limits the practical utility to considerably small image sizes. However, research aimed at enhancing the performance of ZKPs is ongoing, and future technological advancements may enable handling larger image sizes. For instance, Zhang et al. [27] achieved a tenfold acceleration of zk-SNARKs using ASICs. Ma et al. [16] similarly used a graphics processing units to accelerate the proof generation time, achieving up to 48.1 times faster performance compared with traditional methods. Moreover, methods to simplify computational processes have been proposed, such as the “folding” method. This method compresses the propositions being proved [28]. As speed enhancements are being progressively studied, memory consumption will also likely be optimized. This will potentially allow handling of larger image sizes in the future.

Furthermore, we found that our proposal method can handle data sizes approximately 10 KB. Although directly applying our proposal to realistic image data (ranging from several MBs to dozens of MBs) is challenging, splitting data into chunks by modifying the encryption and embedded strings might make the application feasible.

Moreover, our implementations requires a value based on the size of the original data to be processed (encrypted) as an argument during circuit generation. Therefore, a circuit must be generated for each data. The circuit generation time (build time) increases considerably with image data size; for instance, even in ZK-optimized implementation, generating a circuit for a 120×120 image size requires more than 112 min (6,754,928 ms). However, once the circuit is generated, the proof generation time under the same conditions is short, approximately 97 s (96,580 ms). In other words, once a circuit is generated, proof generation is not time intensive. This fact does not pose any practical issues in cases wherein the same image is distributed to various people.

In ZK-friendly implementations, encrypted data is inputted as a public input. Handling encryption keys for images requires careful consideration. Data managed in private IPFS are encrypted. However, if encryption keys are leaked, the encrypted data could be decrypted. Therefore, specific users managing private IPFS should become administrators to carefully manage the keys or a consortium-type blockchain could be established on the same network to set and manage access rights appropriately.

9. Conclusions

A new method was proposed herein to distribute data stored in private IPFS to external entities while making its authenticity verifiable. The method applied a type of ZKP, zk-SNARKs, to verify the CID of data and embed the recipient’s name. This approach enables external entities to verify that the received data are generated from the original data in private IPFS without requiring details such as IPFS access rights and encryption keys.

A standard implementation using conventional cryptographic techniques and a ZK-optimized implementation using ZK-friendly cryptographic schemes were implemented to enhance the computational efficiency of the proposed method. Experiments with a sample program confirmed the effectiveness of the proposed method for an image data size of up to 120×120 pixels.

This proposed method extends the usable range of decentralized storage systems to a hybrid case—distributing internal data to specific external entities as necessary. This study paves a new way for sharing sensitive information across different sectors within and outside a group. However, for the wide practical applicability of the proposed method to larger and more diverse data types, such as images and videos, processing speed must

be improved and data splitting methods must be used, which are within the scope of our future studies.

Author Contributions: Conceptualization, K.S., C.H., T.T.C. and W.O.; writing—original draft preparation, K.S.; writing—review and editing, K.S. and K.I.; supervision, G.M.; project administration, K.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was conducted as a collaborative research project between the University of Tokyo and Callisto Inc., funded by Callisto Inc. This work has been supported by Endowed Chair for Blockchain Innovation and the Mohammed bin Salman Center for Future Science and Technology for Saudi-Japan Vision 2030 (MbSC2030) at The University of Tokyo.

Data Availability Statement: The source code used for the simulations is available on GitHub. https://github.com/blockchaininnovation/circom_image_processing (accessed on 21 March 2024).

Conflicts of Interest: Author Changhee Han, Tsz Tat Chu and Wataru Ozaki were employed by the company Callisto Inc. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

- Groth, J. On the size of pairing-based non-interactive arguments. In Proceedings of the Advances in Cryptology—EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, 8–12 May 2016; Proceedings, Part II 35; Springer: Berlin/Heidelberg, Germany, 2016; pp. 305–326.
- Kumar, S.; Bharti, A.K.; Amin, R. Decentralized secure storage of medical records using Blockchain and IPFS: A comparative analysis with future directions. *Secur. Priv.* **2021**, *4*, e162. [CrossRef]
- Azbeq, K.; Ouchetto, O.; Andaloussi, S.J. BlockMedCare: A healthcare system based on IoT, Blockchain and IPFS for data management security. *Egypt. Inform. J.* **2022**, *23*, 329–343. [CrossRef]
- Hossan, M.S.; Khatun, M.L.; Rahman, S.; Reno, S.; Ahmed, M. Securing ride-sharing service using IPFS and hyperledger based on private blockchain. In Proceedings of the 2021 24th International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 18–20 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
- Lin, Y.; Zhang, C. A method for protecting private data in IPFS. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 5–7 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 404–409.
- Battah, A.A.; Madine, M.M.; Alzaabi, H.; Yaqoob, I.; Salah, K.; Jayaraman, R. Blockchain-based multi-party authorization for accessing IPFS encrypted data. *IEEE Access* **2020**, *8*, 196813–196825. [CrossRef]
- Huang, H.S.; Chang, T.S.; Wu, J.Y. A secure file sharing system based on IPFS and blockchain. In Proceedings of the 2nd International Electronics Communication Conference, Singapore, 8–10 July 2020; pp. 96–100.
- Sun, J.; Yao, X.; Wang, S.; Wu, Y. Blockchain-based secure storage and access scheme for electronic medical records in IPFS. *IEEE Access* **2020**, *8*, 59389–59401. [CrossRef]
- Kang, P.; Yang, W.; Zheng, J. Blockchain private file storage-sharing method based on IPFS. *Sensors* **2022**, *22*, 5100. [CrossRef] [PubMed]
- Uddin, M.N.; Hasnat, A.H.M.A.; Nasrin, S.; Alam, M.S.; Yousuf, M.A. Secure file sharing system using blockchain, ipfs and pki technologies. In Proceedings of the 2021 5th International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 17–19 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–5.
- Li, W.; Guo, H.; Nejad, M.; Shen, C.C. Privacy-preserving traffic management: A blockchain and zero-knowledge proof inspired approach. *IEEE Access* **2020**, *8*, 181733–181743. [CrossRef]
- Bellés-Muñoz, M.; Isabel, M.; Muñoz-Tapia, J.L.; Rubio, A.; Baylina, J. Circom: A circuit description language for building zero-knowledge applications. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 4733–4751. [CrossRef]
- Circom Official Website. Available online: <https://iden3.io/circom> (accessed on 24 March 2024).
- Snarkjs Github Repository. Available online: <https://github.com/iden3/snarkjs> (accessed on 4 June 2024).
- ZCash. Available online: <https://z.cash/> (accessed on 12 July 2024).
- Ma, W.; Xiong, Q.; Shi, X.; Ma, X.; Jin, H.; Kuang, H.; Gao, M.; Zhang, Y.; Shen, H.; Hu, W. Gzkip: A gpu accelerated zero-knowledge proof system. In Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, Vancouver BC Canada, 25–29 March 2023; pp. 340–353.
- Han, C.; Shibano, K.; Ozaki, W.; Osaki, K.; Haraguchi, T.; Hirahara, D.; Kimura, S.; Kobayashi, Y.; Mogi, G. All-in-one platform for AI R&D in medical imaging, encompassing data collection, selection, annotation, and pre-processing. In *Proceedings of the Medical Imaging 2024: Imaging Informatics for Healthcare, Research, and Applications*, San Diego, CA, USA, 18–23 February 2024; SPIE: Bellingham, WA, USA, 2024; Volume 12931, pp. 311–315.
- Miano, J. *Compressed Image File Formats: Jpeg, png, gif, xbm, bmp*; Addison-Wesley Professional: Boston, MA, USA, 1999.

19. Content Identifiers (CIDs). Available online: <https://docs.ipfs.tech/concepts/content-addressing/#cids-are-not-file-hashes> (accessed on 24 March 2024).
20. circom-chacha20 Github Repository. Available online: <https://github.com/reclaimprotocol/circom-chacha20> (accessed on 24 March 2024).
21. The 8 × 8 dot Japanese Font “Misaki Font”. Available online: <https://littlelimit.net/misaki.htm> (accessed on 11 June 2024). (In Japanese)
22. Grassi, L.; Khovratovich, D.; Rechberger, C.; Roy, A.; Schafnegg, M. Poseidon: A new hash function for {Zero-Knowledge} proof systems. In Proceedings of the 30th USENIX Security Symposium (USENIX Security 21), Vancouver, BC, Canada, 11–13 August 2021; pp. 519–535.
23. Circomlib Github Repository. Available online: <https://github.com/iden3/circomlib> (accessed on 21 March 2024).
24. Circomlibjs Github Repository. Available online: <https://github.com/iden3/circomlibjs> (accessed on 21 March 2024).
25. Khovratovich, D. Encryption with Poseidon. 2019. Available online: <https://drive.google.com/file/d/1EVrP3DzoGbmzkRmYnyEDclQcXVU7GI0d/view> (accessed on 19 July 2024).
26. Poseidon-Encryption-Circom2 Github Repository. Available online: <https://github.com/Shigoto-dev19/poseidon-encryption-circom2> (accessed on 21 March 2024).
27. Zhang, Y.; Wang, S.; Zhang, X.; Dong, J.; Mao, X.; Long, F.; Wang, C.; Zhou, D.; Gao, M.; Sun, G. Pipezk: Accelerating zero-knowledge proof with a pipelined architecture. In Proceedings of the 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), Valencia, Spain, 14–18 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 416–428.
28. Kothapalli, A.; Setty, S.; Tzialla, I. Nova: Recursive zero-knowledge arguments from folding schemes. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–18 August 2022; Springer: Cham, Switzerland, 2022; pp. 359–388.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Investigating and Mitigating the Performance–Fairness Tradeoff via Protected-Category Sampling

Gideon Popoola and John Sheppard * 

Gianforte School of Computing, Montana State University, Bozeman, MT 59717, USA;
gideon.popoola@student.montana.edu

* Correspondence: john.sheppard@montana.edu

Abstract: Machine learning algorithms have become common in everyday decision making, and decision-assistance systems are ubiquitous in our everyday lives. Hence, research on the prevention and mitigation of potential bias and unfairness of the predictions made by these algorithms has been increasing in recent years. Most research on fairness and bias mitigation in machine learning often treats each protected variable separately, but in reality, it is possible for one person to belong to multiple protected categories. Hence, in this work, combining a set of protected variables and generating new columns that separate these protected variables into many subcategories was examined. These new subcategories tend to be extremely imbalanced, so bias mitigation was approached as an imbalanced classification problem. Specifically, four new custom sampling methods were developed and investigated to sample these new subcategories. These new sampling methods are referred to as protected-category oversampling, protected-category proportional sampling, protected-category Synthetic Minority Oversampling Technique (PC-SMOTE), and protected-category Adaptive Synthetic Sampling (PC-ADASYN). These sampling methods modify the existing sampling method by focusing their sampling on the new subcategories rather than the class label. The impact of these sampling strategies was then evaluated based on classical performance and fairness in classification settings. Classification performance was measured using accuracy and F1 based on training univariate decision trees, and fairness was measured using equalized odd differences and statistical parity. To evaluate the impact of fairness versus performance, these measures were evaluated against decision tree depth. The results show that the proposed methods were able to determine optimal points, whereby fairness was increased without decreasing performance, thus mitigating any potential performance–fairness tradeoff.

Keywords: fairness; protected categories; machine learning; sampling



Citation: Popoola, G.; Sheppard, J. Investigating and Mitigating the Performance–Fairness Tradeoff via Protected-Category Sampling. *Electronics* **2024**, *13*, 3024. <https://doi.org/10.3390/electronics13153024>

Academic Editors: Niussha Shafiabady and Jianlong Zhou

Received: 11 June 2024

Revised: 23 July 2024

Accepted: 24 July 2024

Published: 31 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As machine learning (ML) algorithms increasingly dominate decision-making and decision-assistance systems, their widespread deployment across various sectors raises pressing issues about the fairness and transparency of their predictions [1]. The potential for these algorithms to perpetuate or exacerbate existing societal biases has propelled a significant body of research to investigate and mitigate algorithmic unfairness. This is critical because the decisions influenced by these algorithms profoundly impact individuals, affecting outcomes in domains ranging from finance and employment to criminal justice and healthcare [2].

The source of unfairness and bias in ML is multifaceted [3]. In particular, it is possible that unfairness arises directly from the ML algorithms themselves due to a possible misalignment of the underlying *inductive bias* of the algorithms vis-à-vis the target concept and data distribution. This is referred to as *algorithmic bias*. An alternative concern lies in potential bias resident in the data used to train the models where, as a direct result of the typical “independent and identically distributed” (IID) assumption employed in most ML

methods, the result of learning is to propagate the bias in predictions such that they match the bias in the underlying data itself. It is this latter situation that constitutes the focus of our work here.

1.1. Bias and Unfairness in Machine Learning

Bias is the prejudicial, unfair, or unequal treatment of an individual or group based on specific features, often referred to as sensitive or protected features [4]. Examples of these protected features include age, race, disability, sex, and gender [5]. Bias in ML can be divided roughly into disparate treatment (direct unfairness) and impact treatment (indirect unfairness) [6]. Direct unfairness happens when protected features are used explicitly in making decisions. Indirect unfairness has become increasingly common today. This type of unfairness does not use protected attributes explicitly; instead, it occurs when reliance on variables associated with these attributes results in significantly different outcomes for the protected groups. These other variables are known as proxy features. Examples of real-world bias include the historical U.S. practice of “redlining”, where home mortgages were denied to residents of zip codes predominantly inhabited by minorities, Amazon hiring process gender bias, Google soap dispenser racial bias, etc. [7].

Though these decision assistance tools help automate the decision-making process, these tools may result in unfair treatment of either individuals or groups, both directly or indirectly [8]. Unfairness can occur in several areas of modeling, such as in the training dataset. This can happen when the training dataset does not provide a fair representation of the protected categories, so the “ground truth” becomes difficult to determine. For example, consider a dataset from a company where a specific group has historically faced discrimination. Specifically, suppose female employees in this company have not been promoted as their male counterparts, who, in contrast, have seen career advancement, despite both groups performing at the same level. In this situation, the true value of female employee contributions—the ground truth—is not visible. As a result, an ML algorithm trained on this data is likely to detect and incorporate this bias, thereby perpetuating existing prejudices. This could lead to the algorithm making discriminatory decisions, such as recommending male candidates for hire or promotion more frequently than equally or more qualified female candidates.

Another area where unfairness can occur is in the ML algorithm itself [9]. ML algorithms can still produce discriminatory decisions, even when trained on an unbiased dataset where the “ground truth” is represented accurately. This situation arises when the system’s errors disproportionately impact individuals from a specific group or minority. For example, consider a breast cancer detection algorithm that exhibits significantly higher false negative rates for Black individuals compared with White individuals, meaning it fails to identify breast cancer more frequently in Black patients than in White patients. If this algorithm is used to inform treatment recommendations, it would erroneously advise against treatment for a greater number of Black individuals than White, leading to racial disparities in healthcare outcomes. This underscores the critical need to ensure that algorithms perform equitably across all groups in terms of their training data and how their errors affect different populations. Results from previous literature have reported several cases of algorithms resulting in unfair treatment, e.g., redlining and racial profiling [10], mortgage discrimination [11], employment and personnel selection [12].

While considerable efforts have been geared toward addressing bias in ML predictions [13,14], much of the existing research has focused on mitigating bias for single protected attributes in isolation [15]. For example, on a dataset with two protected attributes, race, and sex, most existing approaches can learn either a fair model involving race or a fair model involving sex but not a fair model involving both race and sex [7]. However, real-world identities are not singular; they are complex and multifaceted, with individuals often belonging to multiple protected groups simultaneously [16]. For example, an individual can be discriminated against across several protected attributes such as age, race, and sex simultaneously. This intersectionality can lead to compounded forms of bias and

discrimination, which are not adequately addressed by single-variable fairness interventions. Therefore, it is critical to develop methodologies that holistically address personal identities' multidimensional nature. This project seeks to bridge this gap by considering combinations of protected categories, thereby synthesizing these protected categories into comprehensive multicategory groups, and aims to tackle the layered complexities of bias more effectively using novel protected-category sampling methods, thus acknowledging and addressing the multifaceted nature of personal identities and potential biases.

The work presented in this paper is motivated by the problem of using ML algorithms for decision making in socially sensitive areas such as loan assessment, hiring, or mortgage assessment, working with this situation where an individual can belong to several protected categories. Given a labeled training dataset containing two or more protected features, the method proposed combines these protected attributes and then splits them into new multicategories. These new categories are likely to be extremely imbalanced and need to be balanced to improve the fairness of the prediction of our ML algorithms. Popular sampling methods such as over-sampling [17], Synthetic Minority Oversampling Technique (SMOTE) [18], Adaptive Synthetic Sampling (ADASYN) [19], etc., sample data across class labels, which does not align with the goal of our research of sampling across the new multicategories. Hence, a new class of modifications of these sampling methods is proposed that can sample across the new category rather than class labels. This new class of modified sampling is called *protected-category sampling*. The resulting proposed protected-category sampling methods are used to sample and balance the new categories before performing classification. The novelty of this work is two-fold. First, the proposed approach combines the protected categories to form new multicategories that mimic what the identity human being looks like in the real world. The second is the modification of existing sampling methods to conform with the sampling of these new categories in order to make sure that all the new categories have the same number of instances.

For demonstration purposes only, a univariate decision tree was chosen as the classification algorithm. The intent is to demonstrate the effects of the different sampling methods on performance, expecting that similar trends will be exhibited regardless of the underlying learning method. The proposed sampling method was compared with the baseline (unsampled data) using accuracy and F1 as the classification performance metrics, as well as equalized odds differences and statistical parity as the fairness metrics. Also, several analyses were performed to show how maximum depth in the decision tree affects both accuracy and fairness.

1.2. Research Question

Proceeding from empirical observation that a trade-off sometimes exists between fairness and ML performance [20], this research tries to answer several questions, such as how this trade-off might be mitigated. In particular, we seek to answer whether the protected-category sampling method of tackling fairness can mitigate this trade-off. In addition, can we develop a methodological framework that effectively mitigates biases across these combined protected variables without compromising the predictive accuracy of ML models? Finally, we plan to answer the question of how the depth of a decision tree affects both accuracy and fairness metrics, thus exploring the relationship between the level of fit (underfitting through overfitting) and fairness.

1.3. Hypothesis

We hypothesize that, by employing sophisticated protected-category sampling techniques designed for these newly formulated multicategory groups, we can significantly increase model fairness in terms of equalized odds differences without decreasing classification performance in terms of accuracy and F1. Furthermore, we explore the delicate balance between fairness and accuracy, hypothesizing that it is possible to identify strategic points where fairness can be maximized without detrimental impacts on performance. This research challenges existing claims of the existence of trade-offs in fairness and ML

prediction. It sets the stage for future explorations into the multidimensional nature of identity and discrimination in automated decision systems.

1.4. Contributions

The broad problem of fairness in machine learning is significant in that the prevalence of AI and ML systems today is having a major impact on people's lives and livelihoods. While attention to fair ML has increased substantially, there continues to be a need for methods to advance fair ML without negatively impacting ML performance. Based on an in-depth review of the literature and the above need for this type of work, the methods reported here make the following contributions:

1. The commonly-held assumption that there exists an inherent tradeoff between fairness and performance (i.e., accuracy) in machine learning is challenged with evidence provided to support this challenge. In particular, the results in this paper indicate that such a tradeoff can be mitigated, suggesting that any tradeoff is most likely tied to how the data is being managed.
2. Four novel preprocessing methods for sampling data are presented based on applying a multicategory sampling strategy using data captured in protected categories. The methods proceed from the assumption and corresponding hypothesis that balancing the data based on these multicategory properties can increase fairness without adversely affecting machine learning model performance.
3. Experimental results are presented using three datasets studied extensively within the fair ML community. The experiments include comparisons with traditional methods of training with no resampling to demonstrate the relative effects of the proposed methods. The results demonstrate that two of the proposed methods, Protected-Category Synthetic Minority Oversampling Technique (PC-SMOTE) and Protected-Category Adaptive Synthetic sampling (PC-ADASYN), are particularly effective in improving both fairness and performance.
4. A detailed analysis relating the potential effects of underfitting and overfitting on fairness is presented by examining different levels in a decision tree model, with and without using the proposed sampling methods. The results demonstrate the ability of the proposed methods to identify an ideal level of the tree where both fairness and accuracy are maximized.

As a result of the above contributions, this work represents a significant step forward in addressing concerns of fairness in machine learning. A key takeaway from the methods and results reported here is that fairness can be addressed without compromising model performance.

1.5. Organization

This paper is organized as follows. In Section 2, a detailed explanation of fairness and a discussion of several technical fairness metrics are presented. Then, in Section 3, previous literature related to bias mitigation strategies is described. In Section 4, we describe our proposed sampling techniques, dataset, and approach to hyperparameter tuning. In Section 5, we present the results of several experiments along with statistical hypothesis tests as a means of validating these results. In Section 6, the experimental results are discussed, and how each algorithm performs on each dataset and each metric is analyzed. Further results on the impact of tree depth on fairness and accuracy are presented as well. In Section 7, the limitations of this work and corresponding directions for future work are presented, and Section 8 presents a number of conclusions.

2. Background

This study considers fairness when predicting an outcome $y \in \mathcal{Y}$ from a set of features $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ and some additional protected attributes $\mathbf{s} \in \mathcal{S} \subseteq \mathbb{R}^p$, such as race, gender, and sex. For example, in loan prediction, $\mathbf{x}$ represents an applicant's financial history, $\mathbf{s}$ is their self-reported race and gender, and y is whether their loan is approved

or denied. A prediction model is considered fair if its errors are evenly distributed across protected groups like different races or genders. The class predictions from training data $\mathcal{D}$ are denoted as $\hat{Y}_{\mathcal{D}} := h(\mathbf{x}, \mathbf{s})$ for some $h : \mathcal{X} \times \mathcal{S} \rightarrow \mathcal{Y}$ from a class $\mathbf{H}$. The protected attributes $\mathbf{s} \in \mathcal{S}$ in our study are assumed to be binary with a special value n denoting the unprivileged group. For example, $\mathcal{S}$ could be race and n “non-White”; therefore, the binary nature of $\mathcal{S}$ is $\{w, n\}$ where w represents White applicants, who are the privileged group, and n represents non-White applicants, who are the unprivileged group. The definition can be further generalized to nonbinary cases.

Discrimination in labeled datasets can be defined as given a dataset $\mathcal{D}$, feature set $\mathcal{X}$, and protected attribute set $\mathcal{S}$ with domain value $\{w, n\}$. The discrimination in $\mathcal{D}$ with respect to the group $\mathcal{S} = n$ denoted as $dis_{s=n}(\mathcal{D})$ is defined as

$$dis_{s=n}(\mathcal{D}) = \frac{|\{\mathbf{x} \in \mathcal{D} : \mathbf{x}(s) = w, h(\mathbf{x}) = +\}|}{|\{\mathbf{x} \in \mathcal{D} : \mathbf{x}(s) = w\}|} - \frac{|\{\mathbf{x} \in \mathcal{D} : \mathbf{x}(s) = n, h(\mathbf{x}) = +\}|}{|\{\mathbf{x} \in \mathcal{D} : \mathbf{x}(s) = n\}|}$$

The above definition can be translated to the difference in the probability of an applicant being in the positive class for each protected attributes domain $\{w, n\}$. Our study extends the above definition by considering dataset $\mathcal{D}$, which contains two or more protected attributes.

Two popular fairness metrics are used. The first is *equalized odd difference* (EOD), which measures how discriminative or fair our prediction is. EOD states that a binary classifier $\hat{y}$ is fair if its false negative rate (FNR) and true positive rate (TPR) are equal across the domain of $\mathcal{S}$ [21]. FPR and TPR with respect to protected attribute $\mathbf{s} \in \mathcal{S}$ with value n can be defined as

$$TPR_n(\hat{y}) = P(\hat{y} = 1 | y = 1, S = n)$$

$$FPR_n(\hat{y}) = P(\hat{y} = 1 | y = 0, S = n)$$

EOD is then defined mathematically as the difference between TPR and FPR across different groups in a protected attribute. That is,

$$EOD = TPR_n(\hat{y}) - FPR_n(\hat{y}).$$

A fair classifier has an EOD of 0, while an unfair classifier has an EOD of 1. Although achieving a fully fair classifier in practice is almost impossible, this research is geared toward improving EOD without decreasing accuracy. Then, for EOD,

$$FPR_n(\hat{y}) = P(\hat{y} = 1 | y = 0, S = n) = TPR_n(\hat{y}) = P(\hat{y} = 1 | y = 1, S = n)$$

and

$$FPR_w(\hat{y}) = P(\hat{y} = 1 | y = 0, S = w) = TPR_w(\hat{y}) = P(\hat{y} = 1 | y = 0, S = w)$$

To extend the above EOD definition to our multicategory, the EOD is calculated for each column, then the macroaverage of the EOD is presented as the final EOD. The second metric used to measure fairness in ML prediction is *statistical parity* (SP). SP defines fairness as an equal probability of being classified as positive [22]. This can be interpreted as each group in a protected attribute having the same probability of being classified with a positive outcome.

$$P(\hat{y} = 1 | S = w) = P(\hat{y} = 1 | S = n)$$

3. Literature Review

ML algorithms, increasingly utilized for decision making in critical applications such as recidivism, credit scoring, loan decisions, etc., might initially be assumed to be fair and free of inherent bias. However, in reality, they may inherit any bias or discrimination present in the data on which they are trained, as noted by Burt [23]. Moreover, merely removing protected variables from the dataset is insufficient to tackle indirect discrimination and

might, in fact, conceal it. This recognition has heightened the need for more advanced tools, making discovering and preventing discrimination a significant area of research, as highlighted by [24–27].

Bias in ML is a fast-growing topic in the machine learning research community. Bias in an ML model can lead to an unfair treatment of people belonging to certain protected groups. Lately, industrial leaders have started putting more and more emphasis on bias in ML models and software. The Institute for Electrical and Electronics Engineers (IEEE) [28], Microsoft [29], and the European Union [30] have recently published principles for guiding fair AI conduct. These organizations have stated that ML models must be fair in real-world applications. Bias mitigation strategies involve modifying one or more of the following to ensure the predictions made by the ML algorithm are less biased: (a) the training data, (b) the ML algorithm, and (c) the ensuing predictions themselves. These are, respectively, categorized as preprocessing [31], inprocessing [32], and postprocessing approaches [21].

First, the training data can be preprocessed to lower unfairness or bias before training the model. Kamiran and Calders [6] suggest sampling or reweighting the data to neutralize discrimination. This approach can adjust the representation or importance of certain data points to favor (or reduce favor) one class over another. Another method involves changing individual data records directly to reduce discrimination, as explored by [33]. For example, this approach involves altering values in a dataset to decrease identifiable biases against certain groups. Additionally, the concept of *t*-closeness, introduced by Sondeck et al. [34], is applied to discrimination control in the work of [35]. Using *t*-closeness ensures that the distribution of sensitive attributes in any given group is close to the distribution of the attribute in the entire dataset, thereby preserving privacy and preventing discrimination based on sensitive attributes. A common thread among these approaches is balancing discrimination control with the processed data's utility, that is minimizing bias without significantly compromising the data's accuracy, representativeness, and overall usefulness for predictive modeling or analysis. This balance is essential for ensuring that efforts to promote fairness do not inadvertently reduce the quality or applicability of the data.

Overall, the pre-processing method can further be divided into three categories: (1) data modification, (2) data removal, and (3) data resampling. Methods in the first category aim to modify the values of the training data points (including protected attribute values, class values, and feature values) to lower the bias in the dataset. An example of this method is data massaging proposed by [15]. Their approach ranks the training data, and data close to the decision boundary in both privileged and unprivileged groups are flipped. Alternatively, an optimized pre-processing method that learns a probabilistic transformation that edits the classes and features with individual distortion and group fairness was proposed by Fahse et al. [23]. In [36], the original attribute values are replaced with values chosen independently from the class label to train a model roughly achieving equalized odds. Similarly, Peng et al. [37] replace the protected attribute values with values predicted based on other attributes, similar to data imputation.

Methods in the second category aim to train a fair model by removing certain features from the training set. An example of this method is data suppression proposed by Dhar et al. [38]. In their paper, the protected attributes and features that are highly correlated with protected attributes, otherwise known as proxy attributes, are removed from the dataset to train a fair model.

Methods in the third category aim to train a fair model either by adjusting the sample weights or by oversampling the dataset. For example, Krasanakis et al. [39] proposed a reweighting method that iteratively adapts training sample weights with a theoretically grounded model to mitigate the bias–accuracy tradeoff. In [40], Chakraborty et al. proposed FairSMOTE as a method to over-sample training points from minority groups with artificial data points based on Synthetic Minority Oversampling Technique (SMOTE) [18], to achieve balanced class distributions. Also, Yan et al. [41] proposed oversampling the training data from the minority groups with artificial data points to achieve balanced class distribu-

tions. Unlike FairSMOTE, the authors focused on scenarios where protected attributes are unknown and applied a clustering method to identify different demographic groups.

Inprocessing involves methods that modify the way an ML model is trained as a means to reduce bias. In [42], an adversarial debiasing approach was proposed. This approach learns a classifier to increase accuracy and fairness in prediction by including a variable for the group interested by simultaneously learning a predictor and an adversary. This leads to the generation of an unbiased classifier because the predictions do not contain any group discrimination information that the adversary could utilize. Alternatively, an algorithm that takes a fairness metric as part of the loss function and returns a model trained for that fairness metric was proposed in [43]. Kamishima et al. [22] proposed a regularization method, which included a penalty term in the loss function of a classifier to produce an unbiased prediction. Zafar et al. [44] developed a new weighting method whereby they tune the sample weight for each training datum to achieve a specific fairness objective, such as equalized odds on the validation data. Recently, bias mitigation has been approached as a constrained optimization problem by adding a fairness constraint and optimizing the loss to be consistent with that constraint [45,46]. Also, some works modify neural networks by using dropout to drop neurons that belong to protected attributes [47].

Postprocessing methods mitigate bias after fitting an ML model and include approaches such as calibration, constraint optimization, and transformation thresholding [6]. Such methods propose an algorithm that gives favorable outcomes to unprivileged groups and unfavorable outcomes to favorable groups within a given confidence interval around the decision boundary with the highest uncertainty. For example, one approach modifies the peak thresholds of the classifier to yield a specified equal opportunity or equalized odds target. Yet another approach involves randomly mutating the classes of certain predictions into different classes [48].

Several new studies [49,50] combined either preprocessing, inprocessing, or postprocessing to form an ensemble method. For example, Bhaskaruni et al. [50] combine oversampling the imbalance protected class with a decision boundary shifting a postprocessing method to tackle the unfairness problem.

Researchers have delved into various concepts of discrimination and fairness within algorithmic decision making. Disparate impact (referred to previously as indirect fairness), for example, is measured through statistical parity and group fairness, as discussed by Bhaskaruni et al. [50]. On the other hand, the concept of individual fairness, also introduced by Bhaskaruni et al., emphasizes that similar individuals should be treated similarly, regardless of their group affiliation. This approach focuses on fairness at the individual level, ensuring that decisions are made based on relevant attributes rather than group-based stereotypes or biases.

In classifiers and other predictive models, achieving equal error rates across different groups is a key goal, as highlighted by Zhang and Neill [16]. Similarly, ensuring calibration or the absence of predictive bias in the predictions, as discussed by Hardt et al. [21], is crucial. However, the tension between these notions—calibration and equal error rates—is explored by Dwork et al. [51] and Pleiss et al. [52], indicating that simultaneously satisfying both can be challenging. Karimi-Haghighi and Castillo [53] present related work exploring the complexities inherent in achieving algorithmic fairness. Friedler et al. [54] further examines the trade-offs in meeting various algorithmic fairness definitions, especially from a public safety perspective. Given that our work focuses on preprocessing rather than modeling, considerations such as balanced error rates and predictive bias become less directly applicable.

Based on our review of various preprocessing methods, it appears that no work has been conducted attempting to model fairness for two or more protected attributes simultaneously. Also, the sampling method used in prior work focused only on sampling based on class labels rather than the protected categories. Hence, in this paper, preprocessing is emphasized as it represents the most adaptable aspect of the data science pipeline [55]. Preprocessing is distinct in that it does not depend on the choice of modeling algorithm and can be seamlessly

incorporated with data release and publishing mechanisms. This independence and flexibility make preprocessing critical for ensuring data quality and fairness before any analytical or predictive modeling occurs. Finally, we focus on new custom sampling methods that sample the protected category in the data training to build a fair model.

4. Methodology

The focus of our work is to explore sampling methods to enhance fairness in ML without the corresponding prediction performance suffering, thus mitigating the fairness–performance tradeoff. As a result, Four novel sampling methods focused on achieving this goal are proposed. These sampling methods address the imbalanced class problem posed by the new multicategory generated due to the combination of the protected categories. Custom sampling methods are needed because the existing methods sample data based on minority and majority classes, but to mitigate fairness, the new multicategories are sampled to be equal. This, in turn, calls for modifying the existing sampling methods to sample data based on these new categories. This leads to four new sampling methods: protected-category oversampling, protected-category proportional sampling, protected-category SMOTE (PC-SMOTE), and protected-category ADASYN (PC-ADASYN).

4.1. Protected-Category Oversampling

In protected-category oversampling, the first step is to combine the protected categories in the dataset and encode the combination to produce our new multicategory. For example, in the Adult Income dataset, age (young and adult), race (White and others), and sex (male and female) are combined to generate eight new categories, which become ADULTWHITEMALE, ADULTOTHERSMALE, YOUNGWHITEMALE, YOUNGOTHERSMAIL, ADULTWHITEFEMALE, ADULTOTHERSFEMALE, YOUNGWHITEFEMALE, and YOUNGOTHERSFEMALE, respectively. These new categories have varying sample sizes, and the goal of our protected-category oversampling is to balance this new category such that the sample size of each of the new categories matches the size of the category with the highest sample size. To avoid data leakage, the dataset is separated into train and test, applying oversampling only on the training data and then testing on an unsampled test set.

The pseudocode in Algorithm 1 shows our protected category oversampling method in detail. In the algorithm, the largest category was used as the baseline because it is the category with the highest sample size. The sampling process results in new training data with a balanced sample size across the new category. The algorithm works by sampling the rest of the protected categories to match the sample size of the baseline. This sampling is performed by repeating the categories multiple times along with their class labels.

4.2. Protected-Category Proportional Sampling

The Protected-Category Proportional Sampling method is a generalization of protected-category oversampling because the process begins by setting a target sample size (which is a hyperparameter to be tuned, rather than just the size of the largest multicategory), denoted as *targetSamples*. This corresponds to the desired number of instances needed for each category. This target ensures uniformity across all categories, mitigating the risk of model bias towards more frequent categories. The typical result of applying this method is that some categories that have more samples than the *targetSamples* will be under-sampled while others will be oversampled to yield an equal proportion of them in the training dataset. The pseudocode in Algorithm 2 shows the step-by-step of the protected-category proportional sampling method.

Algorithm 1 Protected-Category Oversampling

```

1: baselineCount  $\leftarrow$  sum of entries in 'Largest_Category' of  $X_{train}$ 
2: totalCount  $\leftarrow$  number of entries in  $X_{train}$ 
3: baselineProportion  $\leftarrow$  baselineCount / totalCount
4: balancedData  $\leftarrow$  initialize an empty dataset
5: categories  $\leftarrow$  list of column names in  $X_{train}$  starting with 'combined_category_'
6: for each category in categories do
7:   categoryData  $\leftarrow$  select entries in  $X_{train}$  where category = 1
8:   categoryData  $\leftarrow$  combine categoryData with corresponding labels from  $y_{train}$ 
9:   categoryCount  $\leftarrow$  number of entries in categoryData
10:  targetCount  $\leftarrow$  integer part of totalCount  $\times$  baselineProportion
11:  if categoryCount < targetCount then
12:    sampledData  $\leftarrow$  sample targetCount from categoryData with replacement
13:    balancedData  $\leftarrow$  append sampledData to balancedData
14:  else
15:    balancedData  $\leftarrow$  append categoryData to balancedData
16:  end if
17: end for
18: return balancedData

```

Algorithm 2 Protected-Category Proportional Sampling

```

1: targetSamples  $\leftarrow$  5000
2: sampledBalanced  $\leftarrow$  initialize an empty data set
3: for each column in new_categories.columns do
4:   categoryRows  $\leftarrow$  select rows in new_categories where column = 1
5:   sampledRows  $\leftarrow$  sample targetSamples entries from categoryRows with replacement
6:   for each col in oneHotEncodedBalanced.columns do
7:     sampledRows[col]  $\leftarrow$  0
8:   end for
9:   sampledRows[column]  $\leftarrow$  1
10:  sampledBalanced  $\leftarrow$  append sampledRows to sampledBalanced
11: end for
12: return sampledBalanced

```

4.3. Protected-Category SMOTE

The Protected-Category Synthetic Minority Oversampling Technique (PC-SMOTE) sampling method is a more complex process aimed at mimicking SMOTE but modified for sampling our new categories, rather than class labels. In this approach, the first step was to modify SMOTE to use a fixed number of neighbors and to randomly select one neighbor for the interpolation rather than averaging all of them. The pseudocode in Algorithm 3 shows the procedure for the PC-SMOTE. Since the intent for the method is to use it for the new category sampling, it does not address the generation of class labels directly. Hence, a new function that can generate a new class label for the synthetic data is needed. For this, a new function is defined that generates class labels based on the number of new synthetic data generated and a preselected balance ratio between the two classes. Algorithm 4 shows how our new function generates labels for our synthetic samples. The algorithm first determines the number of samples for each class based on the balance ratio and generates the sample needed for each class. The class labels are the shuffle to prevent algorithmic bias in the classes generated.

These two algorithms are combined together to form PC-SMOTE, as shown in Algorithm 5. In the approach to achieve multicategory balance, each distinct category is iterated over such that the subset of data associated with that category is identified. The number of synthetic samples needed to reach a predefined maximum size per category is then calculated. If additional samples are required, the data is generated using PC-SMOTE,

which interpolates between existing data points and their nearest neighbors. Concurrently, a balanced distribution of synthetic class labels is created with a specified balance ratio by employing Algorithm 4. These synthetic features and labels are then incorporated into the training subset for each category. The process is repeated for all categories, resulting in a balanced dataset. The hyperparameters in this Algorithm 5 are the number of neighbors and balance ratio.

Algorithm 3 Custom Synthetic Minority Oversampling Technique (SMOTE)

```

1: procedure CUSTOMSMOTE(data, n_samples)
2:   syntheticSamples  $\leftarrow$  zero matrix of size (n_samples  $\times$  number of columns in data)
3:   nn  $\leftarrow$  NearestNeighbors(n_neighbors = 7).fit(data)
4:   neighbors  $\leftarrow$  nn.kneighbors(data, return_distance = False)
5:   for i  $\leftarrow$  1 to n_samples do
6:     sampleIdx  $\leftarrow$  random integer from 0 to (number of rows in data − 1)
7:     nnIdx  $\leftarrow$  random choice from neighbors[sampleIdx, 1 :]
8:     diff  $\leftarrow$  data[nnIdx] − data[sampleIdx]
9:     weight  $\leftarrow$  random number from uniform distribution between 0 and 1
10:    syntheticSamples[i]  $\leftarrow$  data[sampleIdx] + weight  $\times$  diff
11:  end for
12:  return syntheticSamples
13: end procedure

```

Algorithm 4 Generate Balanced Synthetic Labels

```

1: procedure GENBALSYNTHLABELS(n_samplesNeeded, balanceRatio)
2:   nClass1  $\leftarrow$  int(n_samplesNeeded  $\times$  balanceRatio)
3:   nClass0  $\leftarrow$  n_samplesNeeded − nClass1
4:   syntheticLabels  $\leftarrow$  [0]  $\times$  nClass0 + [1]  $\times$  nClass1
5:   SHUFFLE(syntheticLabels) ▷ Randomly shuffle the labels
6:   return syntheticLabels
7: end procedure

```

Algorithm 5 Protected-category SMOTE

```

1: balancedDataList  $\leftarrow$  initialize an empty list
2: for each category in categories do
3:   categorySubset  $\leftarrow$  select rows in train_data s.t. 'combined_category' == category
4:   features  $\leftarrow$  remove 'class', 'combined_category' from categorySubset
5:   nSamples  $\leftarrow$  max_size − number of rows in categorySubset
6:   if nSamples > 0 then
7:     syntheticFeatures  $\leftarrow$  PCSmote(features, nSamples)
8:     syntheticLabels  $\leftarrow$  GenBalSynthLabels(nSamples, balanceRatio)
9:     syntheticFeatures['class']  $\leftarrow$  syntheticLabels
10:    syntheticFeatures['combined_category']  $\leftarrow$  category
11:    categorySubsetBalanced  $\leftarrow$  concatenate categorySubset and syntheticFeatures
12:  else
13:    categorySubsetBalanced  $\leftarrow$  categorySubset
14:  end if
15:  append categorySubsetBalanced to balancedDataList
16: end for
17: balancedData  $\leftarrow$  append balancedDataList and reset index

```

4.4. Protected-Category ADASYN

The protected-category ADASYN method mimics adaptive synthetic minority (ADASYN) sampling but is modified slightly to fulfill our goal of protected-category sampling. Our PC-ADASYN algorithm is shown in Algorithm 6. It extends ADASYN by focusing on category

density rather than class imbalance. Specifically, this function operates by finding the nearest neighbors to the data and then calculating the density of each data point's category within its immediate neighborhood. It weights these densities inversely to prioritize minority categories, making it more likely to generate synthetic samples from underrepresented categories. The synthetic samples are created by interpolating between selected data points and their neighbors, similar to SMOTE but using a random weight to vary the interpolation, thus ensuring a diverse synthetic dataset. This approach helps address the imbalance at the category level and enriches the dataset's variance, potentially improving the robustness and fairness of ML models trained on this data. Since this sampling method also generates new samples by interpolating, Algorithm 4 is used to generate class labels for the new synthetic samples.

Algorithm 6 PC-ADASYN for Category-Based Balancing

```

1: procedure PCADASYNCATEGORIES(data, labels, n_samplesNeeded, n_neighbors)
2:   n_neighbors  $\leftarrow$  n_neighbors + 1 ▷ Including the data point itself
3:   nn  $\leftarrow$  NearestNeighbors(n_neighbors).fit(data)
4:   distances, indices  $\leftarrow$  nn.kneighbors(data)
5:   densities  $\leftarrow$  zero array of length(data)
6:   for i  $\leftarrow$  0 to length(data) − 1 do
7:     current_category  $\leftarrow$  labels[i]
8:     neighbor_indices  $\leftarrow$  indices[i][1 :] ▷ Skip the self index
9:     densities[i]  $\leftarrow$  SUM(labels[neighbor_indices] == current_category)
10:  end for
11:  weights  $\leftarrow$  1/(densities + 1) ▷ Add 1 to prevent division by zero
12:  weights  $\leftarrow$  weights/SUM(weights) ▷ Normalize weights
13:  syntheticSamples  $\leftarrow$  empty list
14:  sampleIndices  $\leftarrow$  random choice with replacement from length(data) using weights
15:  for each idx in sampleIndices do
16:    baseIdx  $\leftarrow$  idx
17:    neighborIdx  $\leftarrow$  RANDOMCHOICE(indices[baseIdx][1 :])
18:    diff  $\leftarrow$  data[neighborIdx] − data[baseIdx]
19:    syntheticSample  $\leftarrow$  data[baseIdx] + RANDOM()  $\times$  diff
20:    append syntheticSample to syntheticSamples
21:  end for
22:  return array(syntheticSamples)
23: end procedure

```

Algorithms 4 and 6 are combined to form our protected-category ADASYN sampling, as shown in Algorithm 7. For each category, the data corresponding to that category is isolated and the size deficit relative to the largest category is computed. If additional samples are needed, the PC-ADASYN method is applied, generating synthetic features that respect the category's distribution characteristics. These features are then complemented with synthetically generated labels, maintaining a predefined class balance ratio. The process not only corrects category imbalances but also enriches the dataset, potentially enhancing the predictive accuracy and fairness of models trained on this data.

4.5. Dataset and Hyperparameter Tuning

To test the four sampling methods, a classifier was needed to assess the effects on fairness and performance. Ultimately, the type of classifier is not directly relevant since the goal is to mitigate the fairness–performance tradeoff, rather than to find the best classifier. Therefore, we chose to use univariate decision trees based on CART [56] due to their robustness against noise and missing data. The specific implementation we chose was taken from the sklearn library (version 1.5.1) [57]. In addition, decision trees allow us to control the strength of fit by setting the tree depth of the learned tree. This allows us to compare fairness and performance across different levels of fitting.

The process begins by combining protected categories within each dataset, applying one-hot encoding to create new multicategory features, and then performing label encoding. The datasets were then divided using a stratified 10-fold cross-validation to ensure a representative distribution of classes in each fold. For each fold, training was conducted on sampled data using the previously described methods, while classification was tested on the corresponding unsampled test sets. Consistency in model training was maintained by applying identical tree depth across all sampling methods, and the results provided are averages of the 10-fold runs with their corresponding confidence intervals.

Algorithm 7 Protected-category ADASYN

```

1: balancedDataList ← initialize an empty list
2: for each category in categories do
3:   categorySubset ← select from new_data3 s.t. 'combined_category' == category
4:   features ← remove 'class', 'combined_category' from categorySubset
5:   categoryLabels ← extract 'combined_category' from categorySubset
6:   nSamplesNeeded ← max_size minus number of rows in categorySubset
7:   if nSamplesNeeded > 0 then
8:     syntheticFeatures ← PCAdasyn(features, categoryLabels, nSamplesNeeded)
9:     syntheticLabels ← GenBalSynthLabels(nSamplesNeeded, balanceRatio)
10:    syntheticFeatures['class'] ← syntheticLabels
11:    syntheticFeatures['combined_category'] ← category
12:    categorySubsetBalanced ← concatenate categorySubset with syntheticFeatures
13:   else
14:     categorySubsetBalanced ← categorySubset
15:   end if
16:   append categorySubsetBalanced to balancedDataList
17: end for
18: balancedData ← concatenate balancedDataList and reset index

```

Three datasets were selected from the UCI repository [58] for our analysis: the Adult Income dataset [59], the German Credit [60] dataset, and the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) dataset. The Adult Income dataset aims to predict whether an individual earns above USD 50,000, featuring eight categorical and four numerical attributes, with protected variables corresponding to age (young or adult), sex (male or female), and race (White or others). The adult income dataset was donated to UCI in 1996. The German Credit dataset, used to predict creditworthiness, comprises 20 categorical and two numerical attributes, with protected variables of age and sex. German credit dataset was donated to UCI in 1994. The COMPAS dataset [61], which assesses recidivism rates in the United States, includes six categorical and six numerical features, with protected variables of age, race, and sex. The dataset was published in 2018. These datasets were selected because they represent the state-of-the-art datasets for measuring bias and discrimination and are widely used in other studies on algorithmic bias and fairness (see Section 3). Also, the datasets have various sizes, ranging from small to large, which makes them suitable for testing our sampling methods.

Hyperparameter tuning was conducted using grid search [62] to explore a broad range of parameters, complemented by visual assessments to identify optimal settings that balance Equalized Odds Difference (EOD) and accuracy. For the Adult Income dataset, the optimal hyperparameters included a maximum tree depth of 3 and, for PC-SMOTE and PC-ADASYN, a nearest neighbor setting of 5 with a balanced ratio of 0.34. These parameters were similarly effective for the German Credit dataset. For the COMPAS dataset, a maximum tree depth of 2 was optimal for all sampling methods. PC-SMOTE and PC-ADASYN were adjusted to the nearest neighbor setting of 3 and a balanced ratio of 0.60.

5. Results

The four sampling strategies were applied to the three datasets described above and evaluated their impact using a simple univariate decision tree classifier. The results in Tables 1–3 show notable differences in model performance across five sampling strategies: no sampling, oversampling, proportional sampling, PC-SMOTE, and PC-ADASYN on our three datasets. Each method was assessed based on accuracy, macro F1, Equalized Odds Difference (EOD), and Statistical Parity (SP). The results were measured in accuracy and macro F1 because these two metrics are the most popular classification metrics. Also, limiting the metric to two makes the results comparable for statistical analysis.

To ascertain the statistical significance of each method's results, we used the Friedman test, a nonparametric alternative to the one-way ANOVA with repeated measures. Upon finding significant results from the Friedman test, we proceeded with the Nemenyi post hoc test. This test is used to evaluate pairwise comparisons between the methods to ascertain which methods statistically differ from each other. The Nemenyi test is advantageous in this setting because it accounts for multiple comparisons without assuming normal distributions, thereby providing a robust way to understand specific pairwise differences.

For the Adult Income dataset (Table 1), the no-sampling method yielded an accuracy of 0.82, setting a high baseline for comparison. However, it demonstrated a slightly biased prediction with an EOD of 0.36 and minimal disparity in prediction rates ($SP = 0.02$). In contrast, oversampling maintained the same accuracy but lowered the macro F1 slightly to 0.65, indicating potential overfitting issues while worsening fairness ($EOD = 0.66$) and increasing disparity in prediction rates ($SP = 0.09$). Proportional sampling decreased accuracy to 0.79 but improved the macro F1 to 0.79, suggesting a better balance between precision and recall. However, it significantly increased SP to 0.71, indicating a substantial disparity in positive prediction rates, which raises concerns about the model's fairness. The two custom approaches for SMOTE and ADASYN were designed specifically to improve upon these metrics. PC-SMOTE showed a moderate performance with an accuracy of 0.81 and an improved EOD of 0.25, suggesting enhanced fairness over basic oversampling and the no-sampling method. However, it still recorded lower macro F1 (0.63), indicating misrepresentation issues in synthetic data generation. PC-ADASYN proved to be the most balanced approach, maintaining high accuracy (0.82) and better handling of class imbalances, with a moderate improvement in fairness ($EOD = 0.28$) and a controlled increase in prediction rate disparity ($SP = 0.09$). Overall, the baseline accuracy is statistically significantly better than the proportional sampling while it is not statistically significant as compared with the other sampling methods. For the macro F1 proportional sampling is statistically significantly better than other sampling methods. For the EOD, PC-ADASYN is statistically better than other sampling methods while the no-sampling method SP is statistically significantly better than other sampling methods.

Table 1. Results of the sampling methods on the Adult Income dataset with 95% confidence intervals.

Sampling Method	Accuracy	Macro F1	EOD	SP
No sampling	0.82 ± 0.00	0.66 ± 0.01	0.36 ± 0.18	0.02 ± 0.00
Over-sample	0.82 ± 0.02	0.65 ± 0.06	0.66 ± 0.2	0.09 ± 0.02
Prop. Sample	0.79 ± 0.05	0.79 ± 0.06	0.46 ± 0.12	0.71 ± 0.10
PC-SMOTE	0.81 ± 0.05	0.63 ± 0.07	0.25 ± 0.21	0.07 ± 0.00
PC-ADASYN	0.82 ± 0.04	0.64 ± 0.07	0.28 ± 0.19	0.09 ± 0.02

The results of the experiments on the German Credit dataset also show varying impacts of each sampling strategy, particularly regarding fairness and accuracy, as shown in Table 2. Without sampling, the baseline model achieved an accuracy of 0.72 but exhibited significant bias in its prediction, with an EOD of 0.88, indicating a substantial disparity in error rates between groups in the protected attributes. Implementing oversampling maintained accuracy while improving the macro F1 to 0.68 and notably reducing EOD to 0.37, albeit at

the cost of increased SP to 0.35, highlighting a potential trade-off between different fairness measures. Proportional sampling reduced accuracy slightly to 0.68 but achieved the best F1-score of 0.69. It also lowered EOD to 0.32, suggesting it effectively balances prediction quality with fairness. PC-SMOTE shows an improvement in accuracy with 0.73 but a lower macro F1 of 0.55; the model's fairness shows a huge improvement over the baseline with an EOD of 0.15 and a moderate SP of 0.1. PC-ADASYN shows a similar accuracy to the baseline at 0.72, albeit with the lowest macro F1 of 0.48, suggesting a potential trade-off in precision and recall. However, the model exhibits the best in fairness prediction with an EOD of 0.13 and SP of 0.06. Overall, the result shows that the accuracy of no sampling is not statistically significant to other sampling methods except proportional samplings while for the EOD the results of all the sampling methods are statistically significant in comparison with the no-sampling method.

Table 2. Results of the sampling methods on the German Credit dataset with 95% confidence intervals.

Sampling Method	Accuracy	Macro F1	EOD	SP
No sampling	0.72 ± 0.03	0.62 ± 0.06	0.88 ± 0.10	0.07 ± 0.01
Over-sample	0.72 ± 0.05	0.68 ± 0.04	0.37 ± 0.16	0.35 ± 0.11
Prop. Sample	0.69 ± 0.04	0.69 ± 0.07	0.32 ± 0.26	0.29 ± 0.10
PC-SMOTE	0.73 ± 0.05	0.55 ± 0.09	0.15 ± 0.09	0.1 ± 0.02
PC-ADASYN	0.72 ± 0.03	0.48 ± 0.11	0.13 ± 0.02	0.06 ± 0.00

Table 3 shows the results of our experience with the COMPAS dataset. These results reveal significant variations in model performance across the different sampling strategies. The baseline approach, without sampling, achieved accuracy and a macro F1-score of 0.89 but showed higher disparities in fairness metrics, with an Equalized Odds Difference (EOD) of 0.39 and a Statistical Parity (SP) of 0.29. This underscores potential biases that unadjusted models may exhibit towards protected groups. The application of oversampling slightly improved accuracy to 0.90 but also improved fairness notably, decreasing EOD to 0.26. This suggests effectiveness in reducing outcome disparities without compromising SP. Conversely, proportional sampling, while boosting accuracy and macro F1 to 0.90 and 0.91, respectively, also achieved an EOD of 0.26, improving it over the baseline while also recording a higher SP of 0.36, indicating a potential increase in disparity of positive outcomes across groups. PC-SMOTE and PC-ADASYN, with identical scores in accuracy, macro F1, and SP, managed to maintain fairness improvements with an EOD of 0.30, though these methods also increased SP to 0.47. Overall, the results show that both the accuracy and the EOD of our sampling methods are statistically significantly better than the no-sampling method.

Table 3. Results of the sampling methods on the COMPAS dataset with 95% confidence intervals.

Sampling Method	Accuracy	Macro F1	EOD	SP
No sampling	0.89 ± 0.04	0.89 ± 0.04	0.39 ± 0.15	0.29 ± 0.17
Oversample	0.90 ± 0.03	0.90 ± 0.05	0.26 ± 0.14	0.25 ± 0.07
Prop. Sample	0.90 ± 0.05	0.91 ± 0.04	0.26 ± 0.12	0.36 ± 0.10
PC-SMOTE	0.91 ± 0.02	0.91 ± 0.02	0.30 ± 0.11	0.47 ± 0.19
PC-ADASYN	0.91 ± 0.02	0.91 ± 0.03	0.30 ± 0.13	0.47 ± 0.21

6. Discussion

Results of the experiments on the three datasets substantiate that protected-category sampling can markedly enhance model fairness, often without significantly compromising prediction accuracy. In some cases, improvement in accuracy and macro F1 were also demonstrated. Focusing on the Adult Income dataset results, PC-SMOTE and PC-ADASYN

demonstrated notable improvements in EOD and maintained moderate levels of SP. The efficacy of these methods can largely be attributed to their sophisticated interpolation techniques. For example, a visual examination of the decision trees generated with no sampling and PC-ADASYN provides insightful contrasts. Examples from a single representative fold are shown in Figures 1 and 2, respectively. The decision tree learned without sampling selected its root with a feature closely associated with protected attributes, thus acting as a proxy attribute. This led to pronounced prediction bias as reflected in the EOD. Conversely, the decision tree trained on data generated using PC-ADASYN began with a feature that generalized predictions very well and mitigated bias, as evidenced by a notable enhancement in model fairness and a higher Gini impurity, indicating a purer initial split.



Figure 1. Example decision tree trained on Adult Income with no sampling.

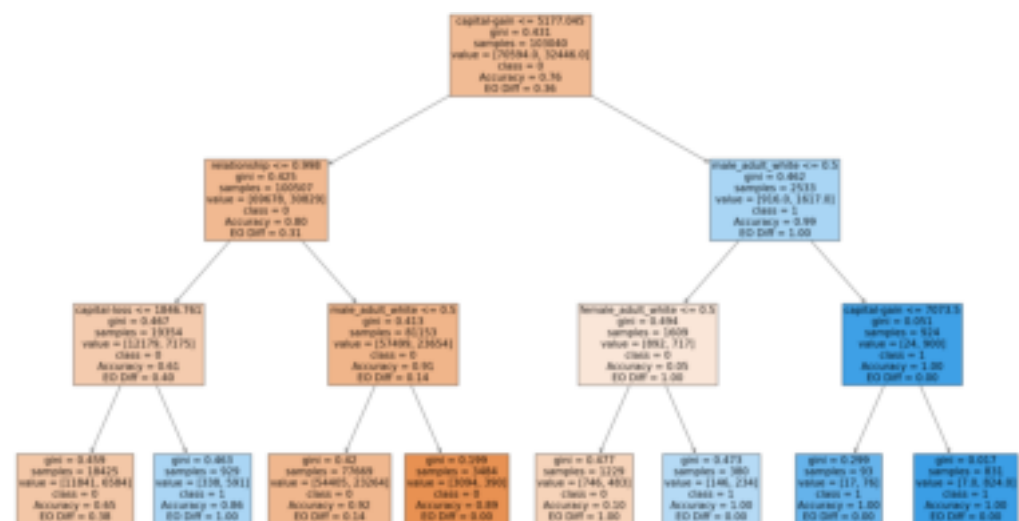


Figure 2. Decision tree of PC-ADASYN on adult income.

6.1. Comparing Fairness vs. Performance

Comparing oversampling and proportional sampling, the methods' approaches to augmenting sample size by duplicating existing data rows were straightforward and did not yield substantial improvements in EOD. This outcome makes sense since these methods tend to replicate existing biases, which can potentially exacerbate fairness issues rather than alleviate them. This is evident, in particular, when considering the Adult Income dataset, where the classes are extremely imbalanced. These naïve replication strategies lack the interpolation capacity of PC-SMOTE and PC-ADASYN to adjust samples near decision boundaries, which is crucial for mitigating the bias in the dataset. In contrast, the interpolation strategies used by SMOTE and ADASYN expand the dataset and enhance its diversity. This is particularly effective for samples near decision boundaries, where slight shifts in the features can affect the fairness of predictions significantly. By interpolating between samples, SMOTE and ADASYN effectively move these boundary samples towards more equitable regions of the feature space, thus directly confronting and reducing bias more effectively than methods that increase sample volume without altering data structure. The class generation function (Algorithm 4) also helps increase the overall class distribution. The results of our PC-SMOTE and PC-ADASYN on Adult income also show superiority over the results obtained in [41], where the accuracy of 0.59 and SP of 0.17 was obtained. Also, the results of PC-SMOTE and PC-ADASYN show superiority over the results obtained in [40], where an EOD of 0.89 and a slightly better accuracy of 0.84.

In examining the results on the German Credit dataset, we observed a trend similar to what was noted in the Adult Income dataset: the no-sampling method has a very high bias regarding EOD. The low SP of 0.07 indicates minimal disparity in the positive prediction rates between the groups, but this in itself is not a good way of measuring fairness since the favored group has more samples than the unfavored one. This has the effect of skewing the calculation of SP since it only counts positive decisions in each group, which are influenced by sample size. One takeaway is the importance of employing multiple fairness metrics to view a model's impact on all stakeholders comprehensively. For oversampling, we saw an improvement in EOD with a similar SP; this shows that increasing the number of samples for each of the multicategory's protected attributes improves the fairness with respect to EOD. In addition, the updated SP reflects what it will look like to have a more equal number of samples for each multicategory, unlike in the baseline where the favored group has five times more samples than the unfavored group. The accuracy of proportional sampling drops because the baseline number of samples selected after hyperparameter tuning was insufficient for the model to generalize the unsampled test set, leading to overfitting. The overfitting was confirmed by considering the training accuracy. Interestingly, the model is not trading accuracy for recall like other models, and this gives proportional sampling the highest macro F1.

Regarding fairness, we found an increase in EOD compared with baseline and oversampling. This arises because each multicategory is represented on the same baseline counts. This can improve the model fairness because the model now has a bigger picture of categories and makes better predictions and ultimately fairer decisions. PC-SMOTE and PC-ADASYN play pivotal roles in significantly reducing bias in model predictions. This consistency confirmed the robustness of these methods across different datasets. Notably, neither method compromises accuracy while both enhance fairness, illustrating their effectiveness in handling the trade-offs typically associated with predictive modeling. These results demonstrate the strong interpolation capabilities inherent in PC-SMOTE and PC-ADASYN. These methods effectively reallocate samples within the feature space, especially moving those in underprivileged regions from negative to more positive decision boundaries. Such adjustments are crucial in mitigating biased outcomes and promoting equity in automated decision-making processes. The macro F1 in both models drops compared with the baseline because a higher number of samples is required for the model to perform better on generalization, which this dataset does not support. Specifically, the dataset has 700 samples for class 0 and 300 samples for class 1, which means the test set

only has 30 samples for class 1. This small number of samples made both models trade recall for precision in class 1. Notably, we saw a low recall for class 1 which ultimately leads to a low f1-score for class 1 and since macro F1 averages the two f1-scores and treats them equally, this affects the performance of both models in macro F1. Overall, the two models yield a fairer model with good accuracy compared with the baseline and other two sampling methods.

The COMPAS dataset's evaluation further validates our sampling methods' effectiveness. The distinct patterns that emerge align with those observed in the Adult Income and German Credit datasets, underscoring the robustness of our findings. Notably, oversampling and proportional sampling techniques have demonstrated substantial improvements in Equalized Odds Difference (EOD) and accuracy, while oversampling also notably improves in SP. This improvement is likely due to the unique composition and balance within the COMPAS dataset, unlike the other datasets in which the classes are imbalanced. The success of oversampling and proportional sampling in this context can be attributed to the balanced nature of the dataset, which allows repeated duplication of existing rows (sampling techniques employed by these methods) to enhance the dataset without introducing a significant skew towards any particular class. This method effectively augments the representation of all classes and the protected attributes in a balanced form, making these techniques particularly effective for datasets where the feature domains contribute equally to predictions and where initial class distributions do not suffer from severe imbalance. This can further be verified from their macro F1 as none of the models is trading precision for recall. The improvement of SP in oversampling can be attributed to the higher number of samples in oversampling in comparison with proportional sampling. Regarding PC-SMOTE and PC-ADASYN, these algorithms show an improvement over baseline in both accuracy and EOD. These trends follow those in the previous results. One notable thing in this results in the large drop in SP which can be attributed to our new label that was generated to make the dataset to be skewed towards the negative class. These results show the difficulty in optimizing for two or more fairness metrics at a time and how this optimization can affect each other.

6.2. Impact of Tree Depth on Fairness and Accuracy

In this study, the impact of decision tree depth on model performance was also investigated, specifically examining how variations in tree depth influence accuracy and EOD. Understanding the depth's effect is crucial as it provides insight into the effects ranging from underfitting to overfitting and helps identify the optimal complexity level at which both accuracy and fairness are maximized. Initially, the decision tree was allowed to grow without constraints to its full depth which on average was about 30 branches. The tree was then examined visually to deduce the maximum depth excluding the nonsplitting branches. To analyze the effects of tree depth systematically, the maximum depth of the trees was allowed to vary from 1 to 30. Each depth limit was evaluated using ten-fold cross-validation to ensure the robustness and generalizability of the findings.

For each configuration of tree depth, the accuracy and EOD were measured on the test set. Additionally, 95% confidence intervals were calculated for the metrics across the ten folds. This statistical analysis highlighted the depth at which the decision tree balanced the trade-off between accuracy and fairness while also considering the underlying statistical bias-variance tradeoff. By doing so, it was possible to pinpoint the "sweet spot"—a delicate point where the decision tree maintains high predictive accuracy without compromising on fairness, effectively countering the often-cited trade-off presented in previous literature. Figures 3–7 show the plots of accuracy and EOD against maximum depth for each of the five sampling methods on the Adult Income dataset.

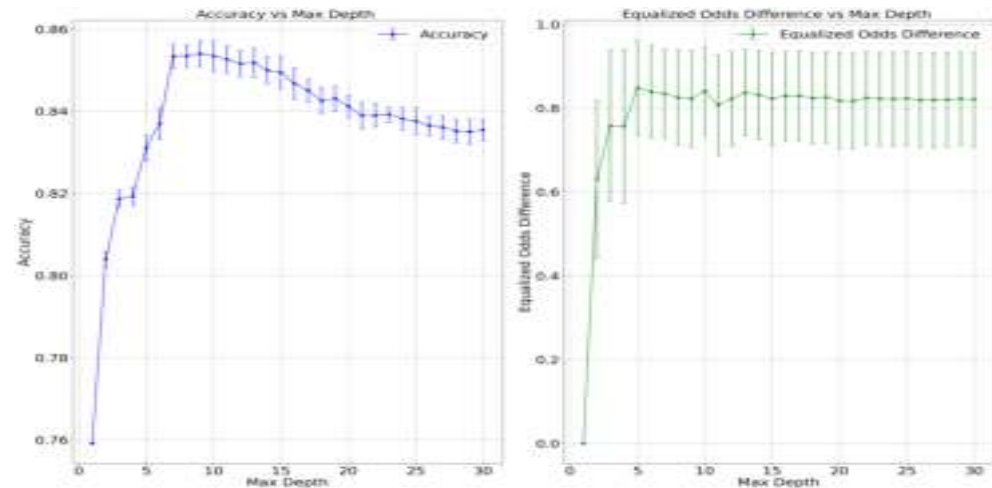


Figure 3. Plots of Adult Income using no sampling, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

Based on results such as those shown in Figures 3 and 4, there is a notable initial increase in accuracy as maximum depth increases for both the no-sampling and the over-sampling methods. However, both methods exhibit a decline in accuracy from a depth of 10 onwards, suggesting the onset of overfitting. Correspondingly, the EOD decreases sharply with increasing depth up to about depth 10, beyond which it stabilizes. This pattern indicates that while deeper trees initially improve fairness, they eventually reach a threshold beyond which no further gains are observed. Recalling that the fairness goal was to minimize EOD, a key observation is that setting the maximum depth between three and five strikes an optimal balance between achieving high accuracy and maintaining low EOD.

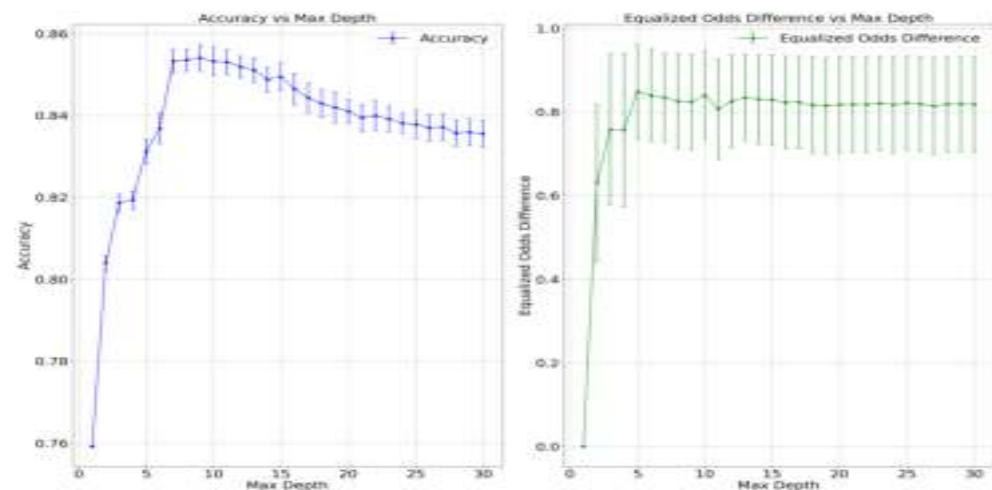


Figure 4. Plots of Adult Income using oversampling, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

When considering the results shown in Figure 5, the proportional sampling method continually increases accuracy with tree depth, peaking at a depth of about 26. Conversely, the EOD initially increases before decreasing and stabilizing at a depth of around 15. The wide confidence intervals observed in the EOD metric suggest significant variability in fairness outcomes. This finding underscores the importance of selecting a depth that minimizes variability in fairness while maximizing accuracy.

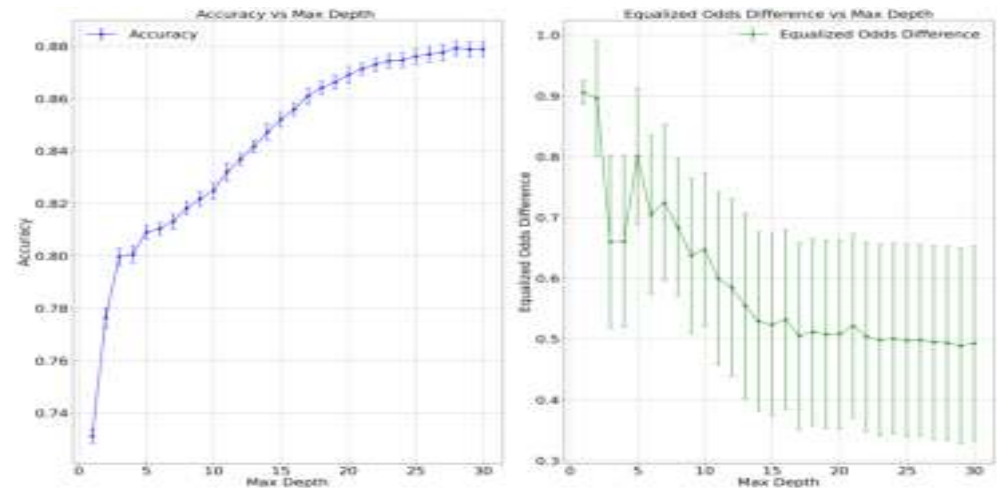


Figure 5. Plots of Adult Income using proportional sampling, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

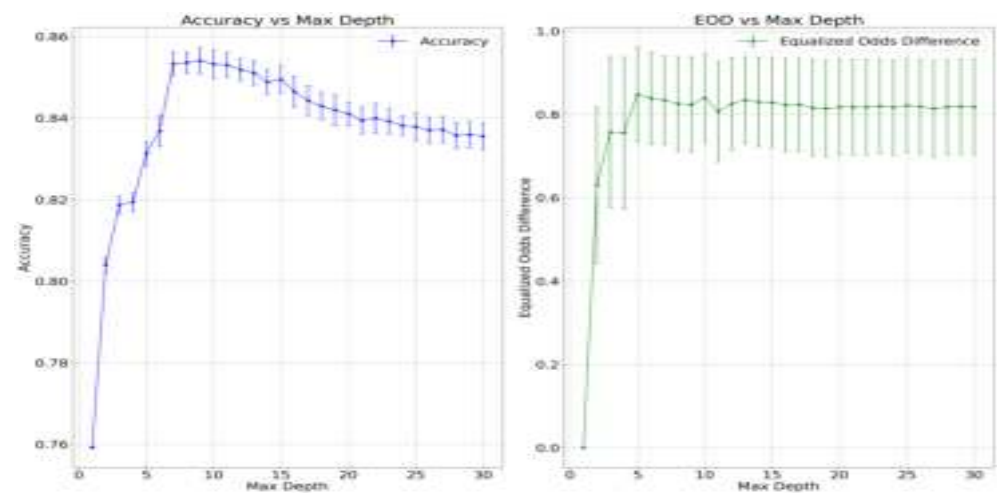


Figure 6. Plots of Adult Income using PC-SMOTE, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

Figures 6 and 7, representing the results using PC-SMOTE and PC-ADASYN, respectively, exhibit slight downward trends in accuracy, which improve briefly before descending again—a pattern indicative of overfitting at greater depths. EOD metrics for these methods show initial stability at lower depths, surge at mid-level depths, and decline, suggesting complex interactions between synthetic sample generation and decision boundary delineation. Given these observations, a maximum depth of 3 was chosen for our experiments, as it represents a “sweet spot” where both accuracy and EOD are optimized.

Given these results, one conclusion is to challenge the often presumed trade-off between accuracy and fairness by demonstrating that our PC-ADASYN method consistently outperforms baselines across all three datasets in terms of both accuracy and fairness. This finding is significant, as it suggests enhancing model fairness without sacrificing accuracy with appropriate sampling methods and model tuning is possible. However, our analysis also reveals scenarios where adjustments to model complexity, specifically the maximum depth of decision trees, can enhance accuracy at the expense of fairness, as indicated by increases in Equalized Odds Difference (EOD). It is expected, however, that coupling sampling methods with inprocessing methods such as fairness-based regularization may offset these effects. These decisions highlight researchers’ discretionary power in balancing model performance metrics depending on their study’s specific objectives and constraints. The quantity of sample and time complexity is like every other sampling method. As the

sample quantity increase, the time complexity increases but overall, the sampling methods have the same time complexity as their underlying algorithms because they have the same functionality.

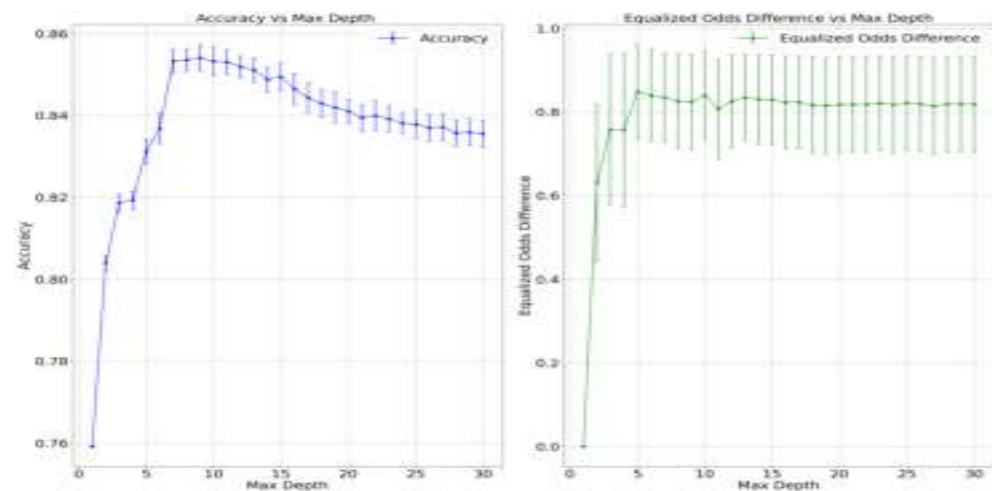


Figure 7. Plots of Adult Income using PC-ADASYN, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

Moreover, our results underscore the complexities of simultaneously optimizing multiple fairness metrics. For instance, efforts to improve Statistical Parity (SP) by favoring more positive predictions for each protected group in the COMPAS dataset led to an inadvertent reduction in negative predictions. This shift adversely impacted the False Positive Rate (FPR), a component of EOD, thereby worsening the EOD metric as SP improved. This phenomenon illustrates the inherent mathematical tensions between fairness metrics, where optimizing one can detrimentally affect another. The COMPAS dataset, with its nearly balanced class distribution, provides a concrete example of how dataset characteristics can influence the behavior of fairness metrics. Optimizing SP in this context implies a skewed measurement of fairness, particularly where inherent differences exist between groups in protected attributes. This is supported by literature indicating that SP may not adequately account for group differences, potentially leading to misleading conclusions about a model's fairness [63].

7. Limitations and Future Work

The very nature of this study is such that it is not possible to address all of the issues surrounding fairness and the so-called fairness–performance tradeoff. As such, there exist limitations in the work reported here. Even so, it is our hope and intent for the work reported here to suggest additional avenues of exploration in this important area.

One limitation of this study is that our sampling method was not specifically designed to optimize for arbitrary fairness metrics. Stated another way, since often inherent tradeoffs exist between the available set of fairness metrics, the decision was made to focus on an approach that was metric agnostic, recognizing that the results could have differed for other metrics. This is also part of the reason why we saw different behaviors between EOD and SP.

In addition, it is acknowledged that, while the underlying ML method should not be relevant to the method proposed, this has not actually been tested. Therefore, in the future, this research will be extended by considering the impacts of other ML algorithms such as logistic regression, fuzzy ID3, K-nearest neighbor, and ensemble methods such as random forests or gradient-boosted trees to assess the generalizability of our new sampling methods. The purpose of such a study would be to verify that our methods are independent of the ML algorithm employed. Furthermore, this would help validate whether the observed improvements in fairness and accuracy are model-specific or can be universally applied.

Additionally, it is acknowledged that only three distinct datasets were considered—datasets that have been studied extensively in the field. This raises a concern that methods are being tailored to these data rather than addressing the broader issue of fairness in ML. To address this, experiments with larger and more diverse datasets are planned to provide deeper insights into the scalability and robustness of our techniques. Another area for future work is to refine our multicategory sampling approach by incorporating more granular subdivisions of protected categories, potentially revealing subtler biases and providing a more nuanced understanding of fairness.

Finally, it is recognized that alternative methods have been proposed for bias mitigation, and these methods have not been studied in this work at all. Future work would entail comparisons with more sampling strategies. A more direct comparison of the proposed methods with inprocessing and postprocessing methods will be conducted. For example, incorporating inprocessing methods, such as regularization [22], or a postprocessing method, such as the Randomized Threshold Optimizer [64], will be explored as possible means to obtain further improvements in both fairness and performance.

8. Conclusions

In this study, the issue of bias in ML predictions was investigated, and a method was developed based on combining protected variables into a new multicategory. In particular, the focus was on the question that has been suggested in the literature of a bias–performance tradeoff and seeking a method to mitigate this tradeoff. The proposed new multicategory approach reflects the multifaceted identity of individuals, acknowledging the complex interplay of attributes that define real-world scenarios. Given the inherent imbalance in this multicategory, four sampling methods tailored to these complex categorizations, rather than traditional class labels, were developed. For purposes of applying a baseline classifier, decision trees were trained, and the effectiveness of these methods was evaluated using three datasets that are often employed in fairness studies. The performance of the methods was compared against baseline methods of no sampling, using accuracy, macro F1, Equalized Odds Difference (EOD), and Statistical Parity (SP) as the evaluation metrics.

The results of the experiments indicate that two of the newly developed sampling techniques—PC-SMOTE and PC-ADASYN—successfully enhance fairness without compromising accuracy. Remarkably, in some cases, these methods also improved accuracy, thus providing evidence counter to the popular claims of a fairness–performance tradeoff. Further analysis of the impact of maximum tree depth on model performance revealed that, while increasing depth initially boosts accuracy, it eventually leads to a decline. Conversely, increasing depth adversely affects fairness, highlighting the challenge of balancing complexity with equity. However, optimal tree depths were identified that simultaneously enhance accuracy and EOD, underscoring the possibility of achieving equity without sacrificing performance.

Ultimately, these findings challenge prevailing notions of an implicit performance–fairness tradeoff within bias mitigation research, suggesting that carefully designed bias mitigation strategies have the ability to sidestep this trade-off. Our approach sets a new precedent for developing more equitable predictive algorithms by redefining how protected attributes are utilized in model training.

Author Contributions: Conceptualization, G.P. and J.S.; formal analysis, G.P. and J.S.; investigation, G.P.; methodology, G.P.; software, G.P.; supervision, J.S.; validation, G.P. and J.S.; visualization, G.P.; writing—original draft, G.P.; writing—review and editing, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets and code implemented in this research work have been uploaded to <https://github.com/horlahsunbo/New-folder> (accessed on 10 June 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Maina, I.W.; Belton, T.D.; Ginzberg, S.; Singh, A.; Johnson, T.J. A decade of studying implicit racial/ethnic bias in healthcare providers using the implicit association test. *Soc. Sci. Med.* **2018**, *199*, 219–229. [CrossRef]
2. Salimi, B.; Rodriguez, L.; Howe, B.; Suci, D. Causal database repair for algorithmic fairness. In Proceedings of the 2019 International Conference on Management of Data, Amsterdam, The Netherlands, 30 June–5 July 2019; pp. 793–810.
3. Kordzadeh, N.; Ghasemaghaei, M. Algorithmic Bias: Review, Synthesis, and Future Research Directions. *Eur. J. Inf. Syst.* **2022**, *31*, 388–409. [CrossRef]
4. Pessach, D.; Shmueli, E. A review on fairness in machine learning. *ACM Comput. Surv.* **2022**, *55*, 1–44. [CrossRef]
5. Aghaei, S.; Azizi, M.J.; Vayanos, P. Learning optimal and fair decision trees for non-discriminative decision-making. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 1418–1426. [CrossRef]
6. Kamiran, F.; Calders, T. Data preprocessing techniques for classification without discrimination. *Knowl. Inf. Syst.* **2012**, *33*, 1–33. [CrossRef]
7. Calmon, F.; Wei, D.; Vinzamuri, B.; Ramamurthy, K.N.; Varshney, K.R. Optimized pre-processing for discrimination prevention. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
8. Shahbazi, N.; Lin, Y.; Asudeh, A.; Jagadish, H.V. Representation bias in data: A survey on identification and resolution techniques. *ACM Comput. Surv.* **2023**, *55*, 1–39. [CrossRef]
9. Chen, Z.; Zhang, J.M.; Sarro, F.; Harman, M. A comprehensive empirical study of bias mitigation methods for machine learning classifiers. *ACM Trans. Softw. Eng. Methodol.* **2023**, *32*, 1–30. [CrossRef]
10. Perzynski, A.; Berg, K.A.; Thomas, C.; Cembali, A.; Smith, T.; Shick, S.; Gunzler, D.; Sehgal, A.R. Racial discrimination and economic factors in redlining of Ohio neighborhoods. *Bois Rev. Soc. Sci. Res. Race* **2023**, *20*, 293–309. [CrossRef]
11. Steil, J.P.; Albright, L.; Rugh, J.S.; Massey, D.S. The social structure of mortgage discrimination. *Hous. Stud.* **2018**, *33*, 759–776. [CrossRef]
12. Salgado, J.F.; Moscoso, S.; García-Izquierdo, A.L.; Anderson, N.R. *Shaping Inclusive Workplaces through Social Dialogue*; Springer: Berlin/Heidelberg, Germany, 2017.
13. Leavy, S.; Meaney, G.; Wade, K.; Greene, D. Mitigating gender bias in machine learning data sets. In Proceedings of the Bias and Social Aspects in Search and Recommendation: First International Workshop, BIAS 2020, Lisbon, Portugal, 14 April 2020; pp. 12–26.
14. Hort, M.; Chen, Z.; Zhang, J.M.; Harman, M.; Sarro, F. Bias mitigation for machine learning classifiers: A comprehensive survey. *ACM J. Responsible Comput.* **2023**, *1*, 1–52. [CrossRef]
15. Fahse, T.; Huber, V.; van Giffen, B. Managing bias in machine learning projects. In *Innovation Through Information Systems: Volume II: A Collection of Latest Research on Technology Issues*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 94–109.
16. Zhang, Z.; Neill, D.B. Identifying Significant Predictive Bias in Classifiers. *arXiv* **2017**, arXiv:1611.08292. [CrossRef]
17. Zheng, Z.; Cai, Y.; Li, Y. Oversampling method for imbalanced classification. *Comput. Inform.* **2015**, *34*, 1017–1037.
18. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
19. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; pp. 1322–1328.
20. Janssen, P.; Sadowski, B.M. Bias in Algorithms: On the trade-off between accuracy and fairness. In Proceedings of the 23rd Biennial Conference of the International Telecommunications Society, Gothenburg, Sweden, 21–23 June 2021.
21. Hardt, M.; Price, E.; Srebro, N. Equality of opportunity in supervised learning. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.
22. Kamishima, T.; Akaho, S.; Asoh, H.; Sakuma, J. Fairness-aware classifier with prejudice remover regularizer. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, 24–28 September 2012; pp. 35–50.
23. Calders, T.; Žliobaitė, I. Why unbiased computational processes can lead to discriminative decision procedures. In *Discrimination and Privacy in the Information Society: Data Mining and Profiling in Large Databases*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 23–33.
24. Burt, A. How to Fight Discrimination in AI. Harvard Business Review. 2020 Available online: <https://hbr.org/2020/08/how-to-fight-discrimination-in-ai> (accessed on 12 July 2024).
25. Siegler, A.; Admussen, W. Discovering Racial Discrimination by the Police. *Northwestern Univ. Law Rev.* **2021**, *115*, 987–1054.
26. Grabowicz, P.; Perello, N.; Mishra, A. How to Train Models that Do Not Propagate Discrimination? Equate and Machine Learning Blog, University of Massachusetts, Amherst. 2022. Available online: <https://groups.cs.umass.edu/equate-ml/2022/04/07/how-to-train-models-that-do-not-propagate-discrimination/> (accessed on 12 July 2024).
27. Khani, F.; Liang, P. From Discrimination in Machine Learning to Discrimination in Law, Part 1: Disparate Treatment. Stanford AI Lab Blog. 2022. Available online: https://ai.stanford.edu/blog/discrimination_in_ML_and_law/ (accessed on 12 July 2024).
28. Shahriari, K.; Shahriari, M. IEEE standard review—Ethically aligned design: A vision for prioritizing human wellbeing with artificial intelligence and autonomous systems. In Proceedings of the IEEE Canada International Humanitarian Technology Conference (IHTC), Toronto, ON, Canada, 21–22 July 2017; pp. 197–201.

29. European Commission; Directorate-General for Communications Networks, Content and Technology. Ethics Guidelines for Trustworthy AI. Publications Office. 2019. Available online: <https://op.europa.eu/en/publication-detail/-/publication/d3988569-0434-11ea-8c1f-01aa75ed71a1/language-en/format-PDF/source-337437547> (accessed on 10 June 2024)
30. Ebers, M.; Hoch, V.R.S.; Rosenkranz, F.; Ruschemeier, H.; Steinrötter, B. The European Commission's Proposal for an Artificial Intelligence Act—A Critical Assessment by Members of the Robotics and AI Law Society (RAILS). *J* **2021**, *4*, 589–603. [\[CrossRef\]](#)
31. Hajian, S.; Domingo-Ferrer, J. A methodology for direct and indirect discrimination prevention in data mining. *IEEE Trans. Knowl. Data Eng.* **2012**, *25*, 1445–1459. [\[CrossRef\]](#)
32. Fish, B.; Kun, J.; Lelkes, Á.D. A confidence-based approach for balancing fairness and accuracy. In Proceedings of the 2016 SIAM International Conference on Data Mining, Miami, FL, USA, 5–7 May 2016; pp. 144–152.
33. Hajian, S. Simultaneous Discrimination Prevention and Privacy Protection in Data Publishing and Mining. *arXiv* **2013**, arXiv:1306.6805. [\[CrossRef\]](#)
34. Sondeck, L.P.; Laurent, M.; Frey, V. The Semantic Discrimination Rate Metric for Privacy Measurements which Questions the Benefit of t -closeness over l -diversity. In Proceedings of the 14th International Conference on Security and Cryptography, Madrid, Spain, 24–26 July 2017; Volume 6, pp. 285–294.
35. Ruggieri, S. Using t -closeness anonymity to control for non-discrimination. *Trans. Data Priv.* **2014**, *2*, 99–129.
36. Romano, Y.; Bates, S.; Candes, E. Achieving equalized odds by resampling sensitive attributes. In Proceedings of the Advances in Neural Information Processing Systems, Online, 6–12 December 2020; pp. 361–371.
37. Peng, K.; Chakraborty, J.; Menzies, T. Fairmask: Better fairness via model-based rebalancing of protected attributes. *IEEE Trans. Softw. Eng.* **2022**, *49*, 2426–2439. [\[CrossRef\]](#)
38. Dhar, P.; Gleason, J.; Roy, A.; Castillo, C.D.; Chellappa, R. PASS: Protected attribute suppression system for mitigating bias in face recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 15087–15096.
39. Krasanakis, E.; Spyromitros-Xioufis, E.; Papadopoulos, S.; Kompatsiaris, Y. Adaptive sensitive reweighting to mitigate bias in fairness-aware classification. In Proceedings of the World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 853–862.
40. Rančić, S.; Radovanović, S.; Delibašić, B. Investigating oversampling techniques for fair machine learning models. In Proceedings of the Decision Support Systems XI: Decision Support Systems, Analytics and Technologies in Response to Global Crisis Management: 7th International Conference on Decision Support System Technology, ICDSS 2021, Loughborough, UK, 26–28 May 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 110–123.
41. Yan, S.; te Kao, H.; Ferrara, E. Fair class balancing: Enhancing model fairness without observing sensitive attributes. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, Ireland, 19–23 October 2010; pp. 1715–1724.
42. HuZhang, B.; Lemoine, B.; Mitchell, M. Mitigating unwanted biases with adversarial learning. In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, New Orleans, LA, USA, 2–3 February 2018; pp. 335–340.
43. Celis, L.E.; Huang, L.; Keswani, V.; Vishno, N.K. Classification with fairness constraints: A meta-algorithm with provable guarantees. In Proceedings of the Conference on Fairness, Accountability, and Transparency, Atlanta, GA, USA, 29–31 January 2019; pp. 319–328.
44. Zafar, M.B.; Valera, I.; Rodriguez, M.G.; Gummadi, K.P. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 1171–1180.
45. Agarwal, A.; Beygelzimer, A.; Dudík, M.; Langford, J.; Wallach, H. A reductions approach to fair classification. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 60–69.
46. Lowy, A.; Baharlouei, S.; Pavan, R.; Razaviyayn, M.; Beirami, A. A Stochastic Optimization Framework for Fair Risk Minimization. *Trans. Mach. Learn. Res.* **2022**. [\[CrossRef\]](#)
47. Spinelli, I.; Scardapane, S.; Hussain, A.; Uncini, A. Fairdrop: Biased edge dropout for enhancing fairness in graph representation learning. *IEEE Trans. Artif. Intell.* **2021**, *3*, 344–354. [\[CrossRef\]](#)
48. Hort, M.; Zhang, J.M.; Sarro, F.; Harman, M. Fairea: A model behaviour mutation approach to benchmarking bias mitigation methods. In Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Athens, Greece, 23–28 August 2012; pp. 994–1006.
49. Bhaskaruni, D.; Hu, H.; Lan, C. Improving Prediction Fairness via Model Ensemble. In Proceedings of the IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, OR, USA, 4–6 November 2019; pp. 1810–1814.
50. Iosifidis, V.; Fetahu, B.; Ntoutsi, E. Fae: A fairness-aware ensemble framework. In Proceedings of the IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 1375–1380.
51. Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; Zemel, R. Fairness through awareness. In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, Cambridge, MA, USA, 8–10 January 2012; pp. 214–226.
52. Pleiss, G.; Raghavan, M.; Wu, F.; Kleinberg, J.; Weinberger, K.Q. On fairness and calibration. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–8 December 2017.
53. Karimi-Haghighi, M.; Castillo, C. Enhancing a recidivism prediction tool with machine learning: Effectiveness and algorithmic fairness. In Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, São Paulo, Brazil, 21–25 June 2017; pp. 210–214.

54. Friedler, S.A.; Scheidegger, C.; Venkatasubramanian, S. The (im)possibility of fairness: Different value systems require different mechanisms for fair decision making. *Commun. ACM* **2021**, *64*, 136–143. [CrossRef]
55. García, V.; Sánchez, J.S.; Mollineda, R.A. On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowl.-Based Syst.* **2012**, *25*, 13–21. [CrossRef]
56. Breiman, L. *Classification and Regression Trees*; Routledge: London, UK, 2017.
57. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
58. Kelly, M.; Longjohn, R.; Nottingham, K. The UCI Machine Learning Repository. 2024. Available online: <https://archive.ics.uci.edu> (accessed on 10 June 2024).
59. Becker, B.; Kohavi, R. Adult. UCI Machine Learning Repository. 1996. Available online: <https://archive.ics.uci.edu/dataset/2/adult> (accessed on 10 June 2024). [CrossRef]
60. Hofmann, H.; Statlog (German Credit Data). UCI Machine Learning Repository. 1994. Available online: <https://archive.ics.uci.edu/dataset/144/statlog+german+credit+data> (accessed on 10 June 2024). [CrossRef]
61. Dressel, J.; Farid, H. The accuracy, fairness, and limits of predicting recidivism. *Sci. Adv.* **2018**, *4*, eaao5580. [CrossRef]
62. Huang, Q.; Mao, J.; Liu, Y. An improved grid search algorithm of SVR parameters optimization. In Proceedings of the 2012 IEEE 14th International Conference on Communication Technology, Chengdu, China, 9–11 November 2012; pp. 1022–1026.
63. Caton, S.; Haas, C. Fairness in machine learning: A survey. *ACM Comput. Surv.* **2023**, *56*, 1–38. [CrossRef]
64. Alabdulmohsin, I.; Lucic, M. A Near-Optimal Algorithm for Debiasing Trained Machine Learning Models. In Proceedings of the 35th Conference on Neural Information Processing Systems, Online, 6–14 December 2021.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Secure Processing and Distribution of Data Managed on Private InterPlanetary File System Using Zero-Knowledge Proofs

Kyohei Shibano ^{1,*} , Kensuke Ito ¹, Changhee Han ², Tsz Tat Chu ², Wataru Ozaki ² and Gento Mogi ¹¹ Department of Technology Management for Innovation, School of Engineering, The University of Tokyo, Tokyo 113-8656, Japan² Callisto Inc., Tokyo 171-0022, Japan

* Correspondence: shibano@tmi.t.u-tokyo.ac.jp

Abstract: In this study, a new data-sharing method is proposed that uses a private InterPlanetary File System—a decentralized storage system operated within a closed network—to distribute data to external entities while making its authenticity verifiable. Among the two operational modes of IPFS, public and private, this study focuses on the method for using private IPFS. Private IPFS is not open to the general public; although it poses a risk of data tampering when distributing data to external parties, the proposed method ensures the authenticity of the received data. In particular, this method applies a type of zero-knowledge proof, namely, the Groth16 protocol of zk-SNARKs, to ensure that the data corresponds to the content identifier in a private IPFS. Moreover, the recipient's name is embedded into the distributed data to prevent unauthorized secondary distribution. Experiments confirmed the effectiveness of the proposed method for an image data size of up to 120 × 120 pixels. In future studies, the proposed method will be applied to larger and more diverse data types.

Keywords: IPFS; zero-knowledge proof; circom; zk-SNARKs; private IPFS; data distribution; data processing; data security



Citation: Shibano, K.; Ito, K.; Han, C.; Chu, T.T.; Ozaki, W.; Mogi, G. Secure Processing and Distribution of Data Managed on Private InterPlanetary File System Using Zero-Knowledge Proofs. *Electronics* **2024**, *13*, 3025. <https://doi.org/10.3390/electronics13153025>

Academic Editor: Aryya Gangopadhyay

Received: 13 June 2024

Revised: 14 July 2024

Accepted: 20 July 2024

Published: 31 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Decentralized systems are robust because they lack a single point of failure; therefore, they are widely applied across enterprise sectors including cryptocurrency, supply chain management, financial services, and digital identity. To store large-sized data such as images, these systems require storage functions that are inherently decentralized. Blockchain, commonly used in conjunction, typically handles smaller data sizes such as transaction histories and operates as a ledger database. The InterPlanetary File System (IPFS) is a prominent decentralized storage system that stores data across multiple nodes to enhance data availability. The IPFS has two variants: public IPFS, wherein the data can be stored by any user with unrestricted access, and private IPFS, wherein a closed network accessible only within specific organizations or groups is established, offering enhanced privacy and security.

When storing data in IPFS, understanding the differences between public IPFS and private IPFS is crucial. Public IPFS allows anyone to access data, while private IPFS is accessible only within specific organizations or groups, enhancing privacy and security. When storing sensitive information, such as confidential data, in public IPFS, applying an appropriate encryption scheme is vital to ensure data protection. By contrast, private IPFS provides higher security for data storage, because it is accessible only within a closed network.

Particularly for organizations such as corporations or healthcare institutions, storing data in public IPFS, despite using strong encryption technologies, carries inherent risks. Moreover, the potential for data leaks due to operational errors exists persistently in such cases; although data is encrypted, it is exposed to the world, rendering it vulnerable to brute force attacks and other security threats.

Regarding accessibility, public IPFS allows general users to directly access and retrieve data. However, in private IPFS, data must be received from members of the organization or group constituting the network. During this process, if data is tampered, then users may be unable to detect it. Therefore, trusting the intermediaries responsible for handling data transfer in such cases becomes mandatory. To address the aforementioned trust issue, a new method is proposed herein for distributing data stored in a private IPFS to external entities while making its authenticity verifiable. The Groth16 [1] protocol of zk-SNARKs, a type of ZKP, is applied to data stored in a private IPFS to ensure the authenticity of the data. Moreover, the recipient's information is embedded into the distributed data to prevent unauthorized secondary distribution. The proposed method of data sharing is important because it is tailored to the private IPFS case.

The differences in several aspects, including security and accessibility, when storing data in public IPFS and private IPFS within the enterprise domain are summarized in Table 1. This study proposes solutions to the threats associated with private IPFS.

Table 1. Comparison between public and private IPFS in the enterprise domain.

Public IPFS		Private IPFS
Trust Model	Trustless	Requires trust in the operating group
Access Restrictions	Accessible by anyone	Accessible only within the operating group
Data Leakage Risk	Constant risk of leakage due to user error	Low risk of leakage within a closed network
Handling of Confidential Information	Requires proper encryption	Data stored in IPFS does not require high-level encryption itself; there is a trust point when passing data to users
Brute Force Attack Risk	Always present	Low
Data Retrieval Method	Direct access by users	Data received from members of the organization or group
Threats	Requires encryption that prevents decryption by unauthorized users	There is a risk of tampering when transferring data to users

The remainder of this paper is organized as follows. Section 2 presents related prior research. Section 3 outlines the fundamental technologies, i.e., ZKP and zk-SNARKs. Section 4 describes the structure of the proposed method, while Section 5 outlines the potential applications of this method. Section 6 presents the implementation of this method, while Section 7 discusses the experiments performed to verify the effectiveness of the implementation. Section 8 presents a discussion of the experimental results, while Section 9 presents the conclusions of the paper and an outline of future challenges.

2. Related Studies

Existing decentralized systems use IPFS, particularly in combination with blockchain technology. Kumar et al. [2] proposed a method for securely managing medical data by integrating IPFS with a blockchain. Azbeg et al. [3] specifically suggested a system that managed and stored medical data using private IPFS and a permissioned blockchain by employing proxy re-encryption to ensure secure decryption by designated doctors. When a physician receives some patient's data, he/she obtains the re-encrypted data via a hospital. Hossan et al. [4] also proposed a system to securely record information for ride-sharing services using IPFS and a private blockchain.

Focusing on controlling the distribution of data managed by IPFS, Lin et al. [5] proposed a system for protecting private data using improved IPFS combined with a blockchain. This system recorded file metadata and accessed permissions on the blockchain, enabling users to control file sharing. Moreover, the system implemented efficient management features using smart contracts, thereby enhancing data security and management

flexibility. Battah et al. [6] developed a system that used multiparty authentication (MPA), proxy re-encryption, and smart contracts on a blockchain for decentralized access control of encrypted data stored in IPFS. Huang et al. [7] introduced a trusted IPFS proxy to realize access control and group key management for encrypted data stored in IPFS. Sun et al. [8] proposed a system that allowed only individuals with appropriate attributes to decrypt encrypted data stored in IPFS using a ciphertext policy attribute-based encryption system, facilitating efficient medical information management. Kang et al. [9] enabled the distribution of data managed using private IPFS and a private blockchain to external users using named data network (NDN). Furthermore, Uddin et al. [10] proposed a file-sharing system that used IPFS and public key infrastructure (PKI) technology without requiring a trusted third party.

Several studies have used ZKP for data distribution. For instance, Li et al. [11] proposed a privacy-preserving traffic management system that combined noninteractive zero-knowledge range proofs with a blockchain. A prototype using Hyperledger Fabric and Hyperledger Ursa met the data privacy requirements for real-time traffic management.

This study proposes a method for appropriately processing and distributing data managed within private IPFS to users outside the network, thereby offering a different approach than those proposed in previous studies. Some studies have adopted proxy re-encryption as an appropriate method for data storage and distribution in IPFS [3,6]. Using this method, distributed data can be re-encrypted to be decrypted with the recipient's private key. Moreover, when storing data in IPFS, recording the hash value of the pre-encrypted data on the blockchain allows recipients to verify the correctness of their received data after decryption. However, this method cannot handle cases where data is processed, such as embedding the recipient's name into the decrypted data, as in this study.

3. Zero-Knowledge Proof

ZKP is a cryptographic protocol that allows a prover to prove the validity of a proposition to a verifier without disclosing any additional information other than the validity of the proposition. The proposition of this study is that the data provided to an external entity is generated based on a given CID. Our goal is to allow a member of private IPFS (prover) to prove this proposition to an external entity (verifier as the recipient of the data) without disclosing any other important information (such as IPFS access rights and encryption keys).

ZKP, specifically the Groth16 protocol of zk-SNARKs used in this study, begins with a trusted setup where both parties establish public parameters that are crucial for the secure generation and verification of proofs. In the ZKP scheme, we first generate a circuit that describes the process for which a proof is intended. The circuit includes the conditions to be verified such as the existence of a CID. Through the ZKP scheme, cryptographic keys—specifically a proving key and a verification key—are created. These keys are crucial for creating a proof for the circuit and its verification. Using the proving key and input data, the prover generates a proof that reveals the validity of the output data against the conditions specified in the circuit. The verifier then uses the verification key to check the proof and the output data. If the proof is valid, this confirms the integrity of data without exposing any underlying information.

In this study, Groth16 processing is performed using circom [12,13] and snarkjs [14]. The process flow is summarized in Figure 1.

zk-SNARKs is employed owing to its noninteractive nature and efficiency, which are particularly advantageous for systems employing smart contracts owing to low computational costs for verifying the proof. Furthermore, zk-SNARKs is known for its high computational requirements and the need for advanced PC specifications. For instance, in one of the representative applications of zk-SNARKs, Zcash [15], proof generation process takes over half a minute for a single anonymous transaction [16].

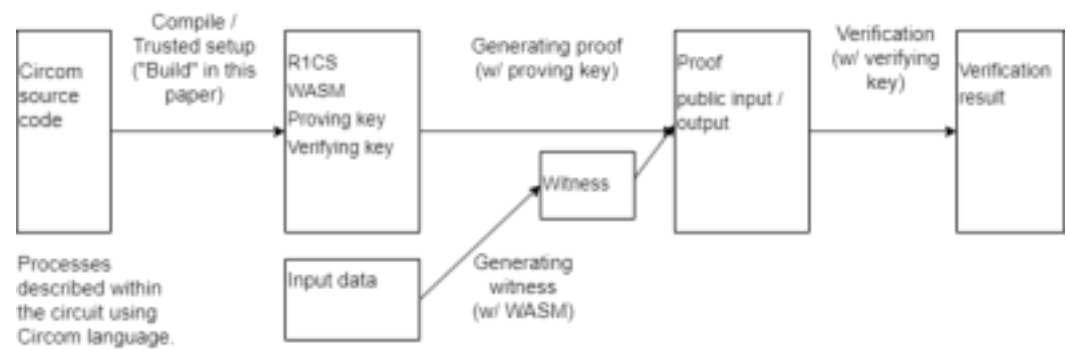


Figure 1. Zero-knowledge proof using circom.

4. Proposed System

Figure 2 presents an overview of the proposed system.

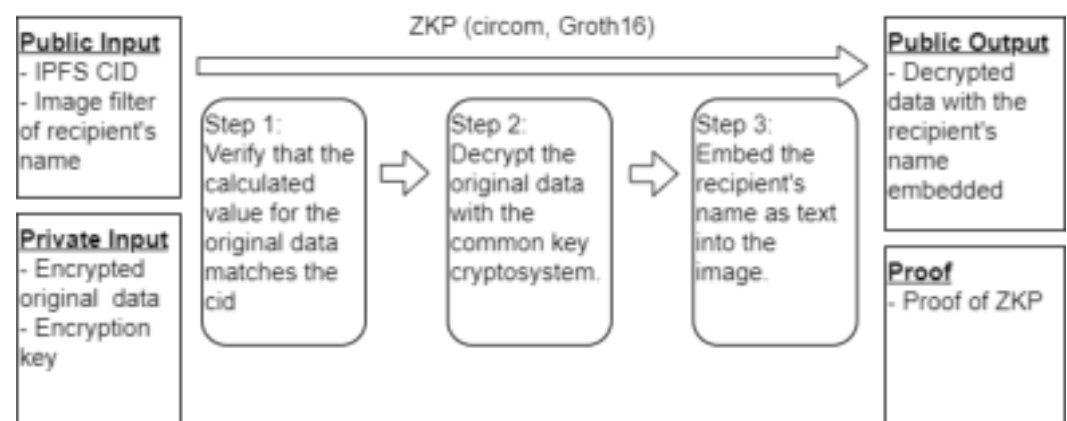


Figure 2. Process flow of ZKP: an example of image data processing.

Herein, we make the following assumptions:

- Private IPFS is operated by a limited number of members.
- Data are stored in IPFS in an encrypted format using symmetric-key cryptography.
- The encryption key is exclusively held by an individual among the members, i.e., an administrator.

The proposed system facilitates the creation of a ZKP proof through the circuit by the encryption key-holding member (equivalent to an administrator). The inputs and outputs (other than proofs) of this process are as follows:

- **Public Input:**
 - CID of the original encrypted data;
 - A filter for embedding the recipient's name in the data;
- **Private Input:**
 - Encrypted original data;
 - Encryption key;
- **Output:**
 - Decrypted data with the recipient's name embedded.

Note that the ZK-optimized implementation (Section 6.1) adds more information to the public input. In particular, the internal process of ZKP involves the following steps:

1. Calculating the CID of the original encrypted data to verify a match with the entered CID.

2. Decrypting the original data using a symmetric-key cryptography.
3. Embedding the recipient's name into the decrypted image raw data based on the filter provided in the public input.

In the proposed system, we do not solely focus on image data but use them as example data to verify the applicability of the proposed scheme. Filters are used to improve the efficiency of processing inside the circuit. As embedding name data inside the circuit is computationally intensive, a considerable portion of the image processing is performed outside the circuit in advance and a filter is created. Using public input and proof, the recipient can verify that the decrypted data (i.e., output) with their name embedded are generated from the original data (contained in the private input) managed with the CID. The recipient can check with at least one member of the network to confirm the existence of the CID in private IPFS.

In summary, the aforementioned process enables the recipient to verify the received data by performing the following tasks:

- verify that the received data were generated from the data managed with the CID of private IPFS,
- confirm using the proof that the entire process was correctly conducted without directly knowing the encrypted data or the encryption key, and
- verify that the CID exists specifically within the private IPFS by asking at least one network member.

The novelty of the proposed system is that it allows data authenticity verification by trusting at least one member of the network even if the recipient do not control the encryption key. (It is natural for the recipient to trust at least one member of a particular multimember system. If none of the members can be trusted, then there will be a marginal incentive to receive data managed by that network).

5. Potential Applications

In the medical industry, patient diagnosis data are managed across multiple medical institutions. Using the proposed system, patients can verify whether the data they receive are indeed managed in private IPFS to ensure the authenticity. An all-in-one platform is also proposed herein for the research and development of machine learning with medical images [17]. On this platform, anonymized medical images are managed in private IPFS operated by a group of medical institutions. The system allows machine learning researchers, who are external to the network, to verify whether the image data are indeed managed in the private IPFS. Moreover, by embedding the information about machine learning researchers in the image data, medical institutions can mitigate the risk of secondary distribution.

If the application is not limited to the embedding of recipient's name, the potential applications of the proposed system can be further expanded. For instance, consider a scenario where a specific company establishes private IPFS for sharing confidential documents among its group companies. If employee data are included, then concealing private data and distributing them to external entities allows these entities to confirm the association of employees with the company while ensuring that their privacy is protected. Furthermore, suppose a university has set up private IPFS to allow only academic staff access to student performance data. In this case, students can verify that their performance data received are genuinely managed in the private IPFS.

Thus, the proposed system supports a hybrid case—distributing internal data to specific external entities as necessary—prevalent in real-world settings.

6. Implementation

The proposed system was implemented to process image data using circom, a renowned tool specialized for constructing zk-SNARKs circuits. Circom enables the description of computational processes within a circuit using its unique language, and the

executable file generated after compilation can be invoked via the JavaScript library, snarkjs. This arrangement allows describing circuit processes in circom, and external processing and circuit correctness testing are performed using JavaScript. For the zk-SNARKs scheme, Groth16 was used; it is known for its relatively faster execution speed than other zk-SNARKs scheme.

In particular, we worked on two types of implementations for image data: a standard implementation using general cryptographic techniques and a ZK-optimized implementation using ZK-friendly cryptographic techniques to reduce the computation time of the circuit. These implementations were used for comparing the required computation times. ZKP circuits require considerably large computation time, even for calculations that can be easily handled by computer software (this is particularly noticeable when dealing with image data). Therefore, computational efficiency is crucial for practical use.

6.1. Standard Implementation

Section 4 describes the data input into the circuit. For simplifying the in-circuit processing, the original encrypted data were formatted as bitmap image data compliant with OS/2 standards. The first 54 bytes of the image data store information such as the width, height, and color depth of images [18]. The color depth is 8 bits and each color component in RGB is allocated one byte, resulting in a representation of 3 bytes per pixel.

Initially, the system checks whether the encrypted data, entered as a private input, matches the CID provided as a public input. If they do not match, the system signifies an error and the image data outputted as the output is a byte sequence where all values are 0x00. CID serves as crucial mechanism for uniquely identifying files and efficiently retrieving data from IPFS. CID has two versions: V0 and V1 [19]. Herein, the more flexible version CID V1 was used. CID includes a hash of the respective data, ensuring different data will have different CIDs. Typically, CID V1 is calculated using the SHA256 hash function, and the standard implementation uses SHA256 to compute CID.

The data structure of CID V1 is as shown in Table 2.

Table 2. CID V1 data structure.

Byte Position	Description	Value in Implementation
First byte	CID version	0x01
Second byte	multibase prefix	0x55: raw data
Third byte	Hash function identifier	0x12: SHA-256
Fourth byte	Hash length	0x20: 32 bytes
From fifth byte	Hash value	SHA-256 hash value (32 bytes)

The encoding for CID is conducted using Base32. Base32 encodes a sequence of bytes constructed based on this structure to generate CID.

Inside the circuit, the entered CID value is decoded from Base32 and the system checks whether the extracted hash value matches the SHA256 hash computed from the encrypted data.

Subsequently, the encrypted data are decrypted. AES-CTR is used as the encryption algorithm, which is a type of symmetric-key cryptography. The AES-CTR encryption and decryption in circom-chacha20 [20] was used. For decrypting AES-CTR encryption, the encryption key and nonce used during encryption are required. They are input into the circuit as a 256-bit key and a 128-bit nonce, respectively, as private inputs. Moreover, AES-CTR handles data volumes in multiples of 16. Therefore, if the length of the image data before encryption is not a multiple of 16, zeros (0x00) are added to the end of the data to align it with this requirement.

Finally, a filter is applied to the decrypted data to embed the recipient's name. Implementing text embedding directly within the circuit can substantially increase the computation load; therefore, a filter is created outside the circuit that performs a considerable

portion of the image processing in advance. The font used for the text representing the recipient's name is the Misaki font [21]. The filter is then used to streamline processing inside the circuit. The filter is a list of numbers where values from 0 to 255 are used to change the color of each pixel in case it differs from that of the pixels in the original image; moreover, a value of 300 indicates the color should remain as in the original image. This filter represents the position on the image where the recipient's name should be inserted. Inside the circuit, the specified pixel colors in the decrypted BMP data are changed based on this filter.

6.2. ZK-Optimized Implementation

ZK-optimized implementation changes the hash function, encryption technology, and in-circuit processing to the standard ZK-friendly encryption implementation. This implementation enhances the computational efficiency and does not evaluate the difference in computation speeds between ZK-friendly encryption and general encryption. Therefore, in-circuit processing was also modified.

Poseidon hash [22] was used as the hash function for computing CID. Notably, using the Poseidon hash for CIDs is not officially supported; therefore, it was developed specifically for this study. Although SHA256 is commonly used in general computations, it demands considerable computation time within ZKP circuits. The Poseidon hash is implemented in circom and JavaScript (circomlib [23] and circomlibjs [24], respectively). It is computed over a finite field with a prime order and can accept up to 16 input variables. The used order is less than the maximum of 32 bytes but greater than the maximum of 31 bytes. This indicates that each of the 16 inputs must contain data not exceeding this order. In this implementation, the data targeted for hash computation are divided into 31-byte segments as input values. If the division exceeds 16 segments, the Poseidon hash is calculated for the first 16 segments. This result is added to the next 15 segments of data for a subsequent Poseidon hash input. The process is repeated until all the input data are used for hash computation. Computationally, if the final input does not complete 16 segments, the missing inputs are set to zero to ensure that the computation always involves 16 inputs.

When generating CID from the Poseidon hash value, the byte sequence should follow the CID V1 data structure and be Base32-encoded. However, to further reduce computation time, this implementation omits the Base32 encoding and directly uses the Poseidon hash value as a substitute for CID. Dividing the input data into 16 segments within the circuit is computationally intensive; therefore, this division is performed outside the circuit and given as an input. In this case, the encrypted data byte sequence and the list of values for calculating the Poseidon hash are provided as public inputs, allowing the verification that both datasets represent the same information. Recipients can confirm that the data being computed for the Poseidon hash and the data being decrypted in the circuit are identical by mutually converting and checking these two values. In this case, as users can obtain the decrypted data, a concern exists regarding password leakage through brute force attacks or other means.

For encryption technology, we adopted Poseidon encryption [25] instead of AES-CTR encryption. Poseidon encryption, implemented in circom and TypeScript (poseidon-encryption-circom2 [26]), involves receiving the public key of the recipient, generating a common key, and ensuring secure encryption and decryption by both parties. In this case, however, a common key is directly generated and used for encryption and decryption. The circuit is provided with two values representing the coordinates of an elliptical curve and a nonce value as private inputs for encryption. Moreover, the filter is implemented in the same manner as in the standard implementation.

7. Evaluation

We created a sample program based on the aforementioned implementations that uses circom to describe the circuit and uses snarkjs for executing the circuit and verifying

proofs. As cryptographic libraries, circom-chacha20 [20], circomlib [23], circomlibjs [24], and poseidon-encryption-circom2 [26] were used.

The standard and ZK-optimized implementations were implemented for each circuit, and their computation times were compared during execution. White bitmap images were the target images, and the experiments were conducted using the letter “A” as the embedded character. As embedding any number of characters does not alter the processing by the filter, embedding a single character allowed for comparing the computation times. Furthermore, we varied the image sizes to measure the execution times for each circuit. The sizes used were 10×10 , 15×15 , 30×15 , 30×30 , 60×30 , 60×60 , 120×60 , 120×120 , and 180×120 pixels. The execution environment was Windows 11 with a Ryzen 9 3950X CPU and 128 GB RAM operating under Ubuntu 22.04 in a WSL2 environment.

Figure 3 shows an example image generated by the circuit, specifically for the 60×30 pixel size using the ZK-optimized implementation. The results for each image size are presented in Table 3, where nonlinear constraints indicate the number of nonlinear constraints in the circuit, build time is the time required to compile circom and output the circuit, and proof gen time is the time required to generate proofs using the circuit. As standard implementation uses AES-CTR encryption, data with 0x00 are appended at the end to ensure that the input size is a multiple of 16.

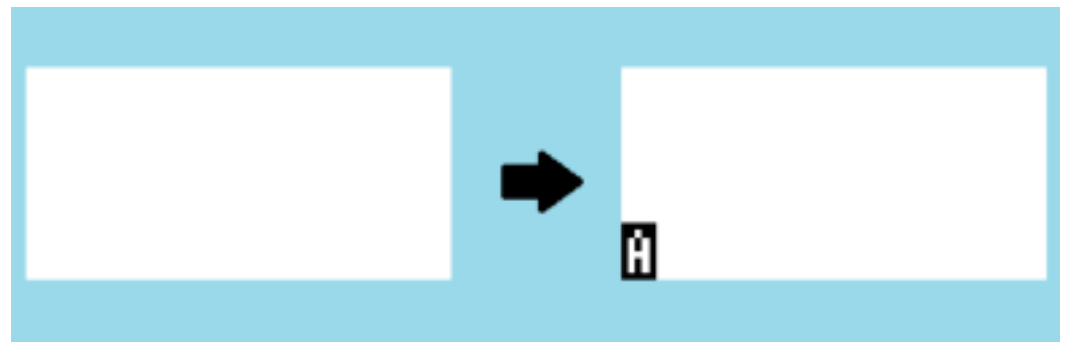


Figure 3. An image generated by the circuit for a 60×30 image size by ZK-optimized implementation.

Table 3. Comparison of the execution time of the circuit.

	Pixel	Image Size [Byte]	Nonlinear Constraints	Build Time [ms]	Proof Gen Time [ms]
Standard	10×10	384	558,341	668,220	14,377
	15×15	784	1,095,292	1,316,666	25,545
	30×15	1440	1,980,688	1,696,370	35,161
	30×30	2816	3,867,989	3,657,578	65,785
	60×30	5456	7,453,300	7,864,583	126,262
ZK-optimized	10×10	376	35,407	128,979	3076
	15×15	775	72,725	173,210	4032
	30×15	1435	134,663	275,630	5900
	30×30	2815	263,450	470,872	9733
	60×30	5455	509,375	850,612	17,571
	60×60	10,855	1,013,483	1,257,418	29,044
	120×120	43,255	4,036,913	6,754,928	96,580

In standard and ZK-optimized implementations for 60×60 pixel and 180×120 pixel image sizes, the system ran out of memory and the computation could not be completed. In ZK-optimized implementation, the number of nonlinear constraints was reduced to approximately one-tenth that of the standard implementation for the same image size.

This reduced the build and proof generation times. However, the maximum manageable image size was still only up to 120×120 pixels, which is considerably small for practical applications.

8. Discussion

Although ZK-friendly cryptographic technologies were used and in-circuit processes were optimized during ZK-optimized implementation, the maximum manageable image size was approximately 120×120 pixels. This limits the practical utility to considerably small image sizes. However, research aimed at enhancing the performance of ZKPs is ongoing, and future technological advancements may enable handling larger image sizes. For instance, Zhang et al. [27] achieved a tenfold acceleration of zk-SNARKs using ASICs. Ma et al. [16] similarly used a graphics processing units to accelerate the proof generation time, achieving up to 48.1 times faster performance compared with traditional methods. Moreover, methods to simplify computational processes have been proposed, such as the “folding” method. This method compresses the propositions being proved [28]. As speed enhancements are being progressively studied, memory consumption will also likely be optimized. This will potentially allow handling of larger image sizes in the future.

Furthermore, we found that our proposal method can handle data sizes approximately 10 KB. Although directly applying our proposal to realistic image data (ranging from several MBs to dozens of MBs) is challenging, splitting data into chunks by modifying the encryption and embedded strings might make the application feasible.

Moreover, our implementations requires a value based on the size of the original data to be processed (encrypted) as an argument during circuit generation. Therefore, a circuit must be generated for each data. The circuit generation time (build time) increases considerably with image data size; for instance, even in ZK-optimized implementation, generating a circuit for a 120×120 image size requires more than 112 min (6,754,928 ms). However, once the circuit is generated, the proof generation time under the same conditions is short, approximately 97 s (96,580 ms). In other words, once a circuit is generated, proof generation is not time intensive. This fact does not pose any practical issues in cases wherein the same image is distributed to various people.

In ZK-friendly implementations, encrypted data is inputted as a public input. Handling encryption keys for images requires careful consideration. Data managed in private IPFS are encrypted. However, if encryption keys are leaked, the encrypted data could be decrypted. Therefore, specific users managing private IPFS should become administrators to carefully manage the keys or a consortium-type blockchain could be established on the same network to set and manage access rights appropriately.

9. Conclusions

A new method was proposed herein to distribute data stored in private IPFS to external entities while making its authenticity verifiable. The method applied a type of ZKP, zk-SNARKs, to verify the CID of data and embed the recipient’s name. This approach enables external entities to verify that the received data are generated from the original data in private IPFS without requiring details such as IPFS access rights and encryption keys.

A standard implementation using conventional cryptographic techniques and a ZK-optimized implementation using ZK-friendly cryptographic schemes were implemented to enhance the computational efficiency of the proposed method. Experiments with a sample program confirmed the effectiveness of the proposed method for an image data size of up to 120×120 pixels.

This proposed method extends the usable range of decentralized storage systems to a hybrid case—distributing internal data to specific external entities as necessary. This study paves a new way for sharing sensitive information across different sectors within and outside a group. However, for the wide practical applicability of the proposed method to larger and more diverse data types, such as images and videos, processing speed must

be improved and data splitting methods must be used, which are within the scope of our future studies.

Author Contributions: Conceptualization, K.S., C.H., T.T.C. and W.O.; writing—original draft preparation, K.S.; writing—review and editing, K.S. and K.I.; supervision, G.M.; project administration, K.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was conducted as a collaborative research project between the University of Tokyo and Callisto Inc., funded by Callisto Inc. This work has been supported by Endowed Chair for Blockchain Innovation and the Mohammed bin Salman Center for Future Science and Technology for Saudi-Japan Vision 2030 (MbSC2030) at The University of Tokyo.

Data Availability Statement: The source code used for the simulations is available on GitHub. https://github.com/blockchaininnovation/circom_image_processing (accessed on 21 March 2024).

Conflicts of Interest: Author Changhee Han, Tsz Tat Chu and Wataru Ozaki were employed by the company Callisto Inc. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

- Groth, J. On the size of pairing-based non-interactive arguments. In Proceedings of the Advances in Cryptology—EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, 8–12 May 2016; Proceedings, Part II 35; Springer: Berlin/Heidelberg, Germany, 2016; pp. 305–326.
- Kumar, S.; Bharti, A.K.; Amin, R. Decentralized secure storage of medical records using Blockchain and IPFS: A comparative analysis with future directions. *Secur. Priv.* **2021**, *4*, e162. [CrossRef]
- Azbeq, K.; Ouchetto, O.; Andaloussi, S.J. BlockMedCare: A healthcare system based on IoT, Blockchain and IPFS for data management security. *Egypt. Inform. J.* **2022**, *23*, 329–343. [CrossRef]
- Hossan, M.S.; Khatun, M.L.; Rahman, S.; Reno, S.; Ahmed, M. Securing ride-sharing service using IPFS and hyperledger based on private blockchain. In Proceedings of the 2021 24th International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 18–20 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
- Lin, Y.; Zhang, C. A method for protecting private data in IPFS. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 5–7 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 404–409.
- Battah, A.A.; Madine, M.M.; Alzaabi, H.; Yaqoob, I.; Salah, K.; Jayaraman, R. Blockchain-based multi-party authorization for accessing IPFS encrypted data. *IEEE Access* **2020**, *8*, 196813–196825. [CrossRef]
- Huang, H.S.; Chang, T.S.; Wu, J.Y. A secure file sharing system based on IPFS and blockchain. In Proceedings of the 2nd International Electronics Communication Conference, Singapore, 8–10 July 2020; pp. 96–100.
- Sun, J.; Yao, X.; Wang, S.; Wu, Y. Blockchain-based secure storage and access scheme for electronic medical records in IPFS. *IEEE Access* **2020**, *8*, 59389–59401. [CrossRef]
- Kang, P.; Yang, W.; Zheng, J. Blockchain private file storage-sharing method based on IPFS. *Sensors* **2022**, *22*, 5100. [CrossRef] [PubMed]
- Uddin, M.N.; Hasnat, A.H.M.A.; Nasrin, S.; Alam, M.S.; Yousuf, M.A. Secure file sharing system using blockchain, ipfs and pki technologies. In Proceedings of the 2021 5th International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 17–19 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–5.
- Li, W.; Guo, H.; Nejad, M.; Shen, C.C. Privacy-preserving traffic management: A blockchain and zero-knowledge proof inspired approach. *IEEE Access* **2020**, *8*, 181733–181743. [CrossRef]
- Bellés-Muñoz, M.; Isabel, M.; Muñoz-Tapia, J.L.; Rubio, A.; Baylina, J. Circom: A circuit description language for building zero-knowledge applications. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 4733–4751. [CrossRef]
- Circom Official Website. Available online: <https://iden3.io/circom> (accessed on 24 March 2024).
- Snarkjs Github Repository. Available online: <https://github.com/iden3/snarkjs> (accessed on 4 June 2024).
- ZCash. Available online: <https://z.cash/> (accessed on 12 July 2024).
- Ma, W.; Xiong, Q.; Shi, X.; Ma, X.; Jin, H.; Kuang, H.; Gao, M.; Zhang, Y.; Shen, H.; Hu, W. Gzkip: A gpu accelerated zero-knowledge proof system. In Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, Vancouver BC Canada, 25–29 March 2023; pp. 340–353.
- Han, C.; Shibano, K.; Ozaki, W.; Osaki, K.; Haraguchi, T.; Hirahara, D.; Kimura, S.; Kobayashi, Y.; Mogi, G. All-in-one platform for AI R&D in medical imaging, encompassing data collection, selection, annotation, and pre-processing. In *Proceedings of the Medical Imaging 2024: Imaging Informatics for Healthcare, Research, and Applications*, San Diego, CA, USA, 18–23 February 2024; SPIE: Bellingham, WA, USA, 2024; Volume 12931, pp. 311–315.
- Miano, J. *Compressed Image File Formats: Jpeg, png, gif, xbm, bmp*; Addison-Wesley Professional: Boston, MA, USA, 1999.

19. Content Identifiers (CIDs). Available online: <https://docs.ipfs.tech/concepts/content-addressing/#cids-are-not-file-hashes> (accessed on 24 March 2024).
20. circom-chacha20 Github Repository. Available online: <https://github.com/reclaimprotocol/circom-chacha20> (accessed on 24 March 2024).
21. The 8 × 8 dot Japanese Font “Misaki Font”. Available online: <https://littlelimit.net/misaki.htm> (accessed on 11 June 2024). (In Japanese)
22. Grassi, L.; Khovratovich, D.; Rechberger, C.; Roy, A.; Schafnegg, M. Poseidon: A new hash function for {Zero-Knowledge} proof systems. In Proceedings of the 30th USENIX Security Symposium (USENIX Security 21), Vancouver, BC, Canada, 11–13 August 2021; pp. 519–535.
23. Circomlib Github Repository. Available online: <https://github.com/iden3/circomlib> (accessed on 21 March 2024).
24. Circomlibjs Github Repository. Available online: <https://github.com/iden3/circomlibjs> (accessed on 21 March 2024).
25. Khovratovich, D. Encryption with Poseidon. 2019. Available online: <https://drive.google.com/file/d/1EVrP3DzoGbmzkRmYnyEDclQcXVU7GI0d/view> (accessed on 19 July 2024).
26. Poseidon-Encryption-Circom2 Github Repository. Available online: <https://github.com/Shigoto-dev19/poseidon-encryption-circom2> (accessed on 21 March 2024).
27. Zhang, Y.; Wang, S.; Zhang, X.; Dong, J.; Mao, X.; Long, F.; Wang, C.; Zhou, D.; Gao, M.; Sun, G. Pipezk: Accelerating zero-knowledge proof with a pipelined architecture. In Proceedings of the 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), Valencia, Spain, 14–18 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 416–428.
28. Kothapalli, A.; Setty, S.; Tzialla, I. Nova: Recursive zero-knowledge arguments from folding schemes. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–18 August 2022; Springer: Cham, Switzerland, 2022; pp. 359–388.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

An Autonomous Cooperative Navigation Approach for Multiple Unmanned Ground Vehicles in a Variable Communication Environment

Xudong Lin  and Mengxing Huang *

School of Information and Communication Engineering, Hainan University, Haikou 570228, China;
linx424523604@163.com

* Correspondence: huangmx09@163.com

Abstract: Robots assist emergency responders by collecting critical information remotely. Deploying multiple cooperative unmanned ground vehicles (UGVs) for a response can reduce the response time, improve situational awareness, and minimize costs. Reliable communication is critical for multiple UGVs for environmental response because multiple robots need to share information for cooperative navigation and data collection. In this work, we investigate a control policy for optimal communication among multiple UGVs and base stations (BSs). A multi-agent deep deterministic policy gradient (MADDPG) algorithm is proposed to update the control policy for the maximum signal-to-interference ratio. The UGVs communicate with both the fixed BSs and a mobile BS. The proposed control policy can navigate the UGVs and mobile BS to optimize communication and signal strength. Finally, a genetic algorithm (GA) is proposed to optimize the hyperparameters of the MADDPG-based training. Simulation results demonstrate the computational efficiency and robustness of the GA-based MADDPG algorithm for the control of multiple UGVs.

Keywords: unmanned ground vehicles (UGVs); genetic algorithm (GA); multi-agent deep deterministic policy gradient (MADDPG); autonomous navigation



Citation: Lin, X.; Huang, M. An Autonomous Cooperative Navigation Approach for Multiple Unmanned Ground Vehicles in a Variable Communication Environment. *Electronics* **2024**, *13*, 3028. <https://doi.org/10.3390/electronics13153028>

Academic Editors: Eric Matson and Young Im Cho

Received: 18 June 2024

Revised: 25 July 2024

Accepted: 30 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A network of distributed unmanned ground vehicles (UGVs) and a central controller is known as a multi-UGV control system [1]. This system enables autonomous domination, autonomous navigation, and autonomous collaboration. It can operate either within a restricted area or as part of a broader transportation system. Multi-UGV control systems offer a unique approach to navigation that is highly reliable, more economical, and conducive to energy savings. In recent years, the urgent demand for multi-UGV navigation systems has encouraged an increasing amount of discussion from academia [2–5].

The navigation of UGVs in a communication environment has been the subject of research [6], and traditional optimization methods have yielded good results [7]. To create an autonomous navigation system, D. Chen et al. [8] developed a heuristic Monte Carlo algorithm that depends on a discrete Hough transform and Monte Carlo localization, which ensures low complexity for processing in real-time. Different from the innovation of algorithms, to perform robustly in unknown and cluttered environments, H. U. Unlu et al. [9] created a robust approach for vision-assisted inertial navigation that can withstand uncertainties. Different from using visual aids, X. Lyu et al. [10] was inspired by a geometric point of view, and they designed a new adaptive sharing factor-integrated navigation information fusion technology scheme that has adaptive navigation in the case of nonlinear systems and uses a non-Gaussian distribution. These traditional optimization methods mentioned above are easy to implement. However, these methods need to be presented with preconditions, which makes them suitable only for static environments. Moreover, in reality, the majority of scenarios involve the collaborative operation of multi-UGVs [11].

Consequently, multi-UGV systems will encounter these two challenges when handling complex scenarios, and it necessitates the incorporation of machine learning (ML) to effectively address them [12–14].

There is a strong rationale for employing ML techniques in UGV navigation, considering the rapid advancements in the field of ML. To achieve improved better-ranging performance, H. Lee et al. [15] provided a ML technique to calculate the distance between the BS and UGVs, which enables localization without any additional infrastructure. Rather than relying on direct ranging, H. T. Nguyen et al. developed a coordination system between unmanned aerial vehicles and UGVs, enabling effective collaborative navigation [15]. However, as the simulation environment becomes more complex, the effectiveness of the proposed solution decreases rapidly. To address this challenge, employing reinforcement learning (RL) algorithms is a promising choice. RL emphasizes how agents can discover the best policy to maximize all rewards when interacting with the environment, which makes it well-suited for exploring and adapting to increasingly complex environments [16].

Research has been driven by discussions on using RL to solve the multi-UGV cooperative navigation issue recently [17]. To avoid collisions with obstacles, X. Huang et al. [18] proposed an innovative deep RL-based UGV local path planning navigation system that leverages multi-modal perception to facilitate policy learning to generate flexible navigation actions. Different from single UGV navigation, to improve the average spectral efficiency, S. Wu et al. [19] proposed trajectory optimization technology based on a joint multi-agent deep deterministic policy gradient (F-MADDPG), which inherits the ability of MADDPG to drive multi-UGVs cooperatively and uses joint averaging to eliminate data isolation and to accelerate convergence. Significant progress has been achieved by these RL-based UGV navigation methods. However, they overlook the limitations of static communication environments and convergence issues arising from the complexity of the environment. These two elements are crucial to take into account while planning cooperative navigation in a communication setting.

Considering the constraints of cooperative communication coverage navigation for UGVs, there are three main challenges to overcome, such as the difficulty of simultaneous control of UGVs, the variation in communication coverage, and the complexity of the cooperative control environment for UGVs. Firstly, traditional control methods such as Q-learning [20], proportional-integral-derivative (PID) control [21], and deep Q-network [22] often yield suboptimal performance in terms of communication coverage when multi-agents require simultaneous control. Secondly, considering the variability in the communication environment during multi-UGV navigation, it is common to encounter areas with poor communication, which hinders effective collaboration among multi-UGVs. However, a promising solution to tackle the challenges of multi-agent cooperative control is offered by multi-agent RL algorithms [23]. These algorithms guide multi-agent collaboration through the centralized training–decentralized execution (CTDE) paradigm [24]. Additionally, in our proposed approach, we introduce a movable UGV BS integrated with the UGVs, allowing for dynamic changes to the fixed communication environment. This collaboration effectively supports the navigation tasks of the UGVs. However, the increased complexity of the constructed environment may pose challenges to algorithm effectiveness and convergence. Fortunately, we mitigate convergence difficulties by adaptive update dynamic hyperparameters using a genetic algorithm (GA) [25]. More fortunately, there has been some research on integrating GA for hyperparameter tuning in RL frameworks. A. Sehgal et al. used a GA to find the hindsight experience replay (HER) used in a deep deterministic policy gradient (DDPG) in a robot manipulation task to help the agent accelerate learning [26]. Different from modifying a single parameter, for the flexible job shop scheduling problem (FJSP), Chen R et al. proposed a GA parameter adjustment method based on Q-learning that changes several key parameters in Q-learning to obtain higher reward values [27]. However, this rewards-based approach is prone to falling into local optimality. Moreover, these methods are not suitable for scenarios where the number of agents increases. To address these issues, Alipour et al. proposed hybridizing a GA with

a multi-agent RL heuristic for solving the traveling salesman problem. In this way, a GA with a novel crossover operator acts as a travel improvement heuristic, while MARL acts as a construction heuristic [28]. Although this approach avoids the risk of local optimality, it abandons the learning process of MARL and only uses it as a heuristic, instead using GA for training, which means that the algorithm will not pay too much attention to the collaboration between intelligent agents. Liu et al. used a decentralized partially observable multi-agent path planning method based on evolutionary RL (MAPPER) to learn effective local planning strategies in mixed dynamic environments. Based on multi-agent reinforcement learning training, they used GA to iteratively extend the originally trained algorithm to a more complex model. Although this method avoids performance degradation in long-term tasks, iterative GA may not necessarily adapt well to more complex environments [29]. In our research, we combine the advantages of the above-mentioned GA papers and adopt the CTDE paradigm to conduct research in a multi-agent RL framework. The GA assigns different weights to algorithm updates based on the transition's contribution, which means that we pay more attention to the hyperparameters that contribute more to model updating rather than those that achieve greater reward values. This allows us to avoid falling into local optimality while increasing the number of agents.

To address these three challenges and achieve cooperative navigation in complex environments, a new multi-UGV communication coverage navigation method is proposed, which is based on a multi-agent deep deterministic policy gradient with GA (GA-MADDPG). The following summarizes the key contributions of the multi-UGV communication coverage navigation method:

- A comprehensive multi-agent pattern is combined into the multi-UGV collaborative navigation system, and the optimal coordination of multi-UGVs within the communication coverage area is formulated as a real-time multi-agent Markov decision process (MDP) model. All UGVs are set as independent agents with self-control capabilities.
- A multi-agent collaborative navigation method with enhanced communication coverage is proposed. By introducing a mobile base station, the communication coverage environment is dynamically changed. Simulation results show that this method effectively improves the communication quality during navigation.
- A GA-based hyperparameter adaptive approach is presented for optimizing UGV communication coverage and navigation. It assigns weights to hyperparameters according to the degree of algorithm updating and makes a choice based on the size of the weight at the next selection, which is different from the traditional fixed-hyperparameter strategy and can escape local optima.

The essay is organized as follows for the remaining portions. The modeling of multi-UGV communication and navigation systems is thoroughly explained in Section 2. The details of the RL method we present is outlined in Section 3. Several experimental comparisons in Section 4 serve to verify the efficacy of our approach. Eventually, we discuss future research directions and summarize the key points of the article in the conclusion Section 5.

2. MDP for Navigation and Communication Coverage for Multi-UGVs in Environments

To emulate the decision-making of multi-UGVs in real-world systems, we adopt an MDP model. With the quick advancement of multi-agent RL, MDP has turned into a trustworthy decision model [30]. In this study, we construct a complex environment with three UGVs and one mobile BS collaborating and which includes various obstacles. Furthermore, we introduce a concept of communication whereby the communication coverage is determined by four fixed BSs and one mobile BS collectively.

2.1. Problem Description

The primary goal of our article is to accomplish multi-agent navigation tasks in a wide range of large-scale, unknown, and complex environments as quickly as possible. The navigation task requires that the UGVs can collaborate according to different environmental characteristics, with the ability to overcome external environmental information

interference, and with the ability to efficiently and autonomously track targets in real-time. More specifically, the extent of communication coverage is collaboratively established by both the stationary BSs and the mobile BS. Within this communication coverage area, the mobile BS and the UGVs engage in cooperative navigation. We construct a task scenario with multiple optimization objectives. The objective of the UGVs is to successfully reach the destination, while the mobile BS is tasked with dynamically adjusting communication coverage in real-time, aiming to optimize the communication quality for the UGVs. The UGVs and mobile BS perform globally optimal cooperative navigation to achieve their respective and common goals.

2.2. Modeling of the Environment

Our work involves simulating a real environment where multi-UGVs collaborate to reach a target point. Additionally, this environment includes obstacles that obstruct the movement of the UGVs, replicating real-world scenarios. We utilize a multi-agent particle environment (MPE) [24] as the base environment for our secondary development, as shown in Figure 1. In this environment, we utilize M UGVs (where M is defined as three), W mobile BSs (where W is defined as one), and a certain number of obstacles. The objective of the UGVs is to collaboratively avoid collisions and reach their respective optimal target points while taking into account communication in the global state. In simpler terms, the UGVs choose an obstacle avoidance route with better communication to coordinate their movement towards the target point (the communication model will be elaborated on in Section 2.3). The task of the mobile BS is to enhance communication for the three movable units by adjusting the communication coverage in the global state, which is exhibited in Figure 1b.

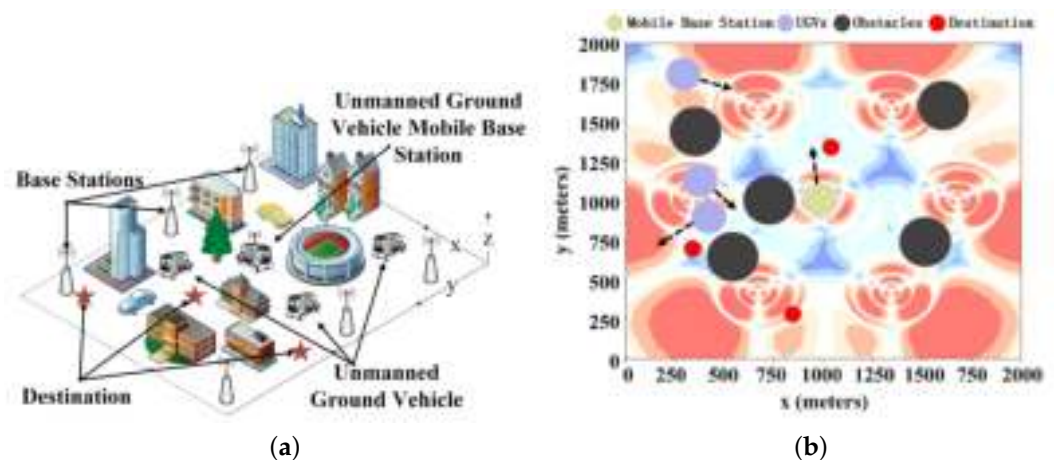


Figure 1. Schematic diagram of the collaboration of a swarm of UGVs in a communication-enabled environment. (a) 3D urban environment. (b) Top view of the visualized communication environment.

2.3. Modeling of the Communication Coverage

In our simulation, we integrated communication into the MPE environment and used it as a criterion to evaluate task completion. In this subsection, we present the communication channel model that we adopted, along with the communication model that is influenced by the movement of the mobile BS, as shown in Figure 1b. The communication area within the middle red circle varies with the location of the movable BS, as illustrated in Figure 2. Note that Figure 2a–l represent diagrams depicting how the communication environment changes with the movement of the mobile BS at step t . The mobile BS is initially positioned in the center in Figure 2a and gradually transitions towards the lower right corner, as depicted in Figure 2l. This relocation of the mobile BS is prompted by its observation of the movement pattern of the UGVs. Consequently, the mobile BS is relocated from the center towards the lower right corner to enhance communication quality in that area, thereby expanding the red coverage zone as shown in Figure 2. Conversely, the relocation of the

mobile BS results in a reduction in the coverage area with superior communication quality in the upper left quadrant. Furthermore, to accentuate the evolving communication quality and enhance the clarity of communication changes, we have delineated the variances between each diagram and its preceding counterpart.

We have constructed a total of M BSs in the environment, where M is defined as seven and includes one mobile BS. The signal power gain obtained by the UGVs from BS m ($m \leq 7$) is defined as p_t^m . Subsequently, the signal-to-interference ratio (SIR) is utilized as the primary criterion for evaluating the communication of the UGVs. This criterion can be expressed as:

$$\text{SIR}_t \triangleq \frac{p_t^{I_t}}{\sum_{m \neq I_t} p_t^m} \quad (1)$$

where $I_t \in \{1, \dots, M\}$ represents the BSs that are not associated with the UGVs at step t . It is worth noting that, for the sake of simplicity, we have omitted the effects of noise, as it is well known that the performance of BS-UGV communication is often constrained by interference. Furthermore, with global frequency reuse, we have taken into account the worst-case situation in which all of these unrelated BSs contribute to the interference term in the Equation (1). In our study, the UGVs received signal power at step t mainly depending on their relative positions to the BSs, and p_t can be written as:

$$p_t = \bar{P} \beta(q_t) G(q_t) \tilde{h}_t \quad (2)$$

where $\bar{P}$ represents the transmit power of the BSs, while $\beta(q_t)$ represents the large-scale channel gain; the large-scale channel gain takes into account the effects of path loss and shadow fading. It can be expressed as:

$$\beta(q_t) = \beta_0 \left(\frac{d_0}{d(q_t)} \right)^\gamma \quad (3)$$

where β_0 is the path loss at the reference distance d_0 , $d(q_t)$ is the distance between the UGV and the BS, and γ is the path loss exponent. And $G(q_t)$ denotes the BS antenna gain; the BS antenna gain considers the directional gain of the UGV relative to the BS antenna. It can be represented by the antenna radiation pattern, which is typically expressed as:

$$G(q_t) = G_{\max} \cdot A(\theta_t, \phi_t) \quad (4)$$

where $G_{\max}$ is the maximum antenna gain, and $A(\theta_t, \phi_t)$ is the gain function of the UGV's position relative to the main lobe direction of the antenna. These parameters typically rely on the location q_t of the UGV. Additionally, the random variable $\tilde{h}_t$ is used to incorporate the effects of fading. It is important to note that each UGV has an independent SIR at each step t , which is utilized to evaluate the communication performance of the UGVs at that specific time. It also should be noted that during the initialization of the scenario, the initial positions of all base stations, including the movable base station, are fixed, i.e., they are all at a fixed position, and then the three UGVs and the movable base station are trained to take different actions through the strategy, at which time, based on the selected action, the next position of the movable base station is determined by the selected action as well as the original position together. The value of q_t is fixed at this point because q_t is only related to the position variable (x, y) . It can be seen that the initial position of the mobile BS is pre-set, while the subsequent q_t is the decision variable and is determined by the action of the mobile BS, which aims to provide a better communication environment to the remaining UGVs.

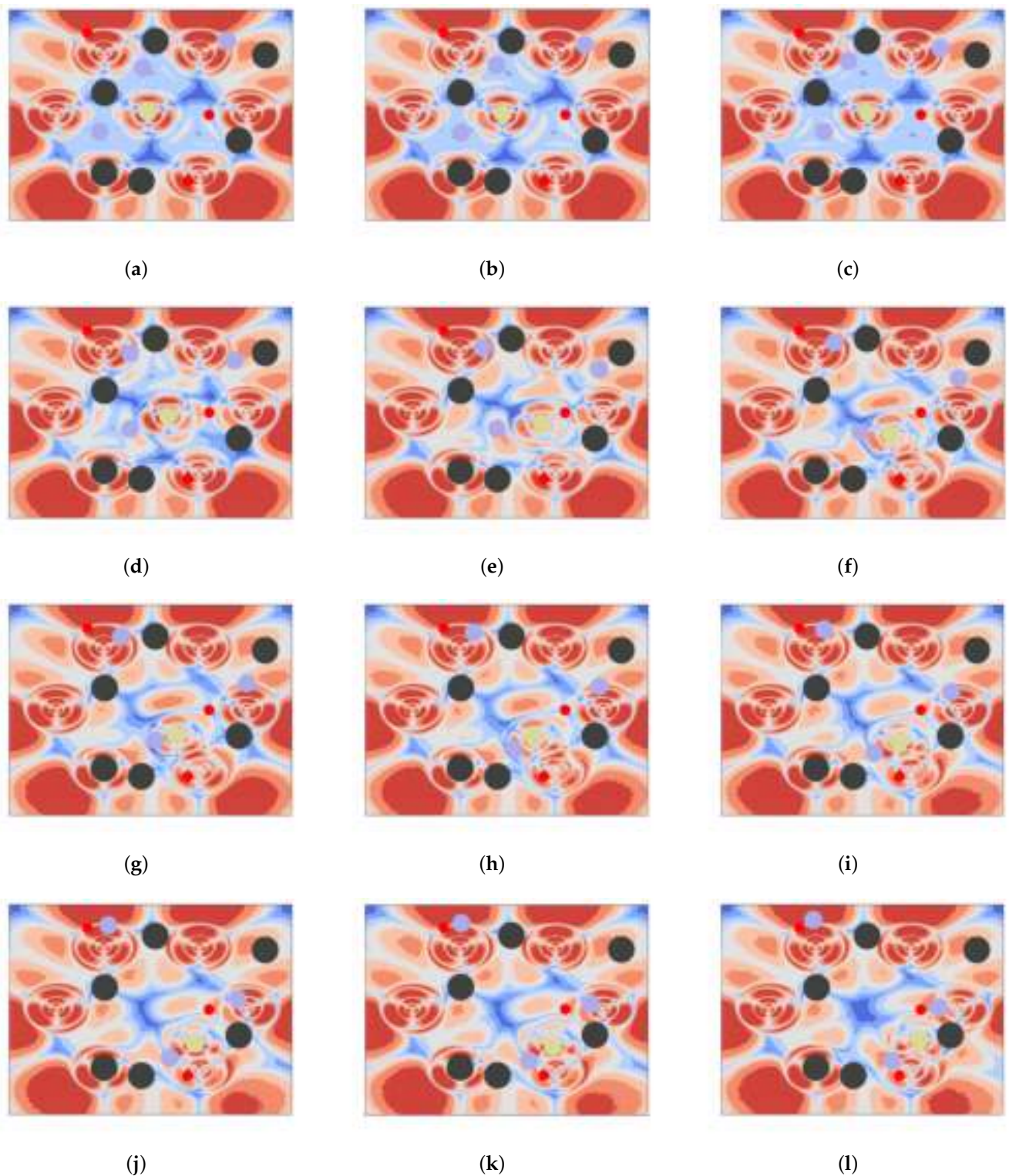


Figure 2. The results of changing communication in the environment as the mobile BS moves. The red areas indicate better communication, whereas the blue areas indicate poorer communication, Brown circles represent the mobile BS, blue circles represent UGVs, black circles represent obstacles, and red circles represent target points. This movement aims to enhance communication quality for the UGVs.

2.4. The State and Action of the UGVs

The state of the UGVs is denoted as $s = (s_1, s_2, \dots, s_N)$. For each UGV u , the state is defined as $s_u = (s_{Pu}, s_{Eu})$, where $s_{Pu} = (x_u, y_u, v_{xu}, v_{yu}, SIR_u)$ is a combination of position (x_u, y_u) , speed (v_{xu}, v_{yu}) , and SIR_u . Additionally, $s_{Eu} = (x_{ug}, y_{ug}, x_0, y_0, v_{ox}, v_{oy})$ represents the data that the UGVs observe other UGVs or obstacles. The term s_u depicts the positions of the agent in a coordinate system. However, in many actual situations, it may not be possible to acquire absolute locations. Therefore, the agent and barriers can be modeled in a polar coordinate system for movement. In our original formulation, $s_{Eu} = (x_{ug}, y_{ug}, x_0, y_0, v_{ox}, v_{oy})$ is intended to represent the observed data for each UGV u . To clarify, (x_{ug}, y_{ug}) represents the distance from the g entity (including the UGVs and all obstacles) to UGV u . And (x_0, y_0) represents the global coordinate position of UGV u . Through a series of transformation calculations, we can also obtain the global positions of other entities observed by UGV u . The combination of these components allows each UGV to navigate toward its goal while considering the presence and motion of obstacles or other UGVs.

The action of UGVs is denoted as $a = (a_1, a_2, \dots, a_N)$, which is defined as a collection of individual actions for each UGV in a multi-agent system. In this particular paper, the motion of UGVs is simplified by assuming an initial velocity of 0 and a constant acceleration, which is represented by a formulation: $v_t = v_0 + at$, which is defined as a 2-dim vector.

2.5. Reward Function

The primary aim is finding the optimal collaborative strategy for a specific state in order to navigate collaboratively during step t and the next step $t + 1$ with improved communication. At step t , the specific state is denoted as s_t . The reward of taking action a_t can be represented by $r(s_t, a_t)$. Consequently, the total reward of adopting policy π can be expressed as:

$$\mathcal{R}(\pi) = L \left[\sum \gamma^t r(s_t, \pi(s_t)) \mid s_0 = s \right] \quad (5)$$

Our objective is to determine the optimal strategy, denoted as π^* , that maximizes the overall reward while adhering to all given constraints. The primary focus of the article is to obtain the policy that yields the highest possible reward, denoted as $R(\pi)$, among all possible policies π .

It should be noted that the navigation principles for the mobile BS are similar to those of the UGVs. Both the UGVs and mobile BS can use similar principles for path planning and obstacle avoidance based on their target positions and current environmental data. The tasks of UGVs are threefold: First, UGVs reach their destination through collaborative navigation. Second, UGVs should try to avoid collisions. Third, UGVs should travel in a communication environment with high quality. The mobile BS has only two tasks: One is to work with the UGVs to adjust the communication coverage by adjusting the position, thus ensuring that the UGVs move within a high-quality communication range. One is to avoid collisions as much as possible, similar to the objective of the UGVs. It is also important to note that the initial position of the mobile BS is at the very center of the scene in all the scenarios we set up and that it co-moves with the UGVs without preempting them. So in this training model, the reward function of the UGVs mainly consists of three parts based on a theoretical foundation. Firstly, it is related to the distance between the UGVs and the target point. Secondly, it is related to the number of collisions, including collisions between UGVs, collisions between UGVs and the mobile BS, and collisions between UGVs and obstacles. Finally, it is related to the SIR obtained by the UGVs at step t , which can be formulated as $r(s_t, a_t)$.

$$r(s_t, a_t) = SIR_t - D(UGVs, target) - coll \quad (6)$$

where SIR_t represents the comprehensive communication quality obtained by all UGVs at each step t , and the definition of SIR_t has been introduced in detail in Equation (1). $D(UGVs, target)$ represents the sum of the lengths between all UGVs and their respective

destinations at each step t ; it should be noted that no UGV has a fixed destination to reach, which means that all UGVs will autonomously allocate the destination to be reached based on their strategies and observations. The term *coll* represents the number of collisions that occurred among all UGVs at each step t .

The calculation formula of SIR_t in Equation (6) has been introduced in Section 2.3. Communication directly impacts the reward function of the UGVs, where higher communication results in a larger reward. Consequently, the UGVs are incentivized to prioritize locations with better communication, encouraging them to move extensively toward those areas.

$D(UGVs, target)$ is computed by:

$$D(UGVs, target) = \sqrt{(x_u - x_{target})^2 + (y_u - y_{target})^2} \quad (7)$$

where (x_u, y_u) contains coordinate information for all the target points, which indicates that the loss also diminishes as the distance between the UGVs and the destination gets smaller. Consequently, a smaller loss corresponds to a higher reward. In essence, the UGVs are more likely to receive a greater reward when they are in closer proximity to the target point.

The term *coll* can be confirmed as:

$$coll = \begin{cases} 0, & \text{if } D(UGVs, tuple) > K \\ -1, & \text{if } D(UGVs, tuple) \leq K \end{cases} \quad (8)$$

where $D(UGVs, tuple)$ represents distances between the UGVs and various entities such as other UGVs, the mobile BS, and obstacles in the given scenario. Additionally, a constant “K” is utilized to assess the possibility of a collision. If the distance between any two entities is less than the value of K, a collision is registered. Consequently, by employing this approach, multi-agents collaborate to minimize the occurrence of collisions.

3. RL Multi-Agent Communication Coverage Navigation with GA

In this section, we describe a concise summary of the MDP formulation for communication coverage navigation with cooperation between the mobile BS and the UGVs. Next, we introduce the DDPG algorithm [31], which is designed for continuous control space. Building upon these foundations, we develop an innovative RL algorithm called GA-MADPPG to address the challenges in communication coverage and navigation. The GA-MADPPG algorithm comprises two main components. Firstly, we adopt the MADPPG algorithm, which extends DDPG following the CTDE paradigm. This allows us to leverage the benefits of MADPPG in handling multi-agent systems and continuous control problems. Secondly, we integrate GA into the MADPPG algorithm, enabling real-time hyperparameter updates based on the loss function during the training process. The proposed policy highlights the GA-MADPPG algorithm’s ability to dynamically adjust hyperparameters based on the loss function. By combining these two components, GA-MADPPG aims to achieve efficient communication coverage and navigation in complex environments.

3.1. MDP Model

The multi-agent Markov game, a significant expansion of the MDP in a multi-agent scenario, is the subject of [32]. In this game, the theoretical state of N agents is represented by s . At each epoch t , the agents keep track of the current state s_t and select an action a_t . Following this, the state enters the following state $s_t + 1$, and all agents are given a reward, $r(s_t, a_t)$.

For the evaluation of action–value functions and state–action mapping value functions, calculating the value function for stochastic policies entails:

$$V_{\pi}(s_t) = \mathbb{E} \left[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l}, a_{t+l}) \mid s_t \right] \quad (9)$$

where the discount factor is $\gamma \in [0, 1)$. And the action–value function is computed as follows:

$$Q_{\pi}(s_t, a_t) = \mathbb{E} \left[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l}, a_{t+l}) \mid s_t, a_t \right] \quad (10)$$

Learning an ideal π^* strategy that optimizes the overall anticipated return is the goal of all agents.

$$\pi^* = \arg \max_{\pi} \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \quad (11)$$

3.2. Fundamentals of the DDPG Approach

DDPG is a deep deterministic policy gradient algorithm developed to tackle continuous action control problems. It is based on policy gradients and directly adjusts the policy parameters θ to optimize the objective function.

$$J(\theta) = \mathbb{E}_{s \sim p^{\pi}, a \sim \pi_{\theta}} \quad (12)$$

which is the core idea behind DDPG, as it involves taking the policy gradient $\nabla_{\theta} J(\theta)$ at each step. The policy gradient can be expressed as follows:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{s \sim p^{\pi}, a \sim \pi_{\theta}} [\nabla_{\theta} \log \pi_{\theta}(a \mid s) Q^{\pi}(s, a)] \quad (13)$$

where $Q^{\pi}(s, a) = \mathbb{E}[R \mid s^t = s, a^t = a]$ is an action–value function, and p^{π} is the state distribution.

Deterministic policies can also be incorporated into the policy gradient framework and are denoted as $\mu_{\theta} : \mathcal{S} \mapsto \mathcal{A}$ [1]. Specifically, under certain circumstances, we can write the gradient of the objective $J(\theta) = \mathbb{E}_{s \sim p^{\mu}}[R(s, a)]$ as follows:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{s \sim \mathcal{D}} \left[\nabla_{\theta} \mu_{\theta}(a \mid s) \nabla_a Q^{\mu}(s, a) \Big|_{a=\mu_{\theta}(s)} \right] \quad (14)$$

The theorem requires the action space a to be continuous, as it depends on $\nabla_a Q^{\mu}(s, a)$.

Deep neural networks are used in the DDPG method, which is a variation of the deterministic policy gradient algorithm, to estimate policy μ and critic Q_{μ} . It is an off-policy approach, meaning it learns from experiences during training. In addition to the online network, DDPG also uses a target network to stabilize training. The target network is periodically revised to mitigate the effects of policy oscillations during learning.

3.3. Multi-Agent Deep Deterministic Policy Gradient

The DDPG policy demonstrates the agent’s inherent robustness and generalization capabilities, leading to maximized performance [31]. This benefit makes DDPG particularly well-suited for learning in challenging circumstances where unknowns and external interference are present. In light of this, we adopted a training paradigm for communication coverage navigation based on the MADDPG. The agent in the environment is autonomous and unable to interact with other agents, yet it is perceptible. At each step t , the agent is unable to observe the current mobility schemes of other agents. The benefit of CTDE is that it eliminates the need to address the trade-offs between agents, and the optimization goal is to increase the total return of all agents [33].

$$G = \langle \hat{s}, a, p, r, o, u \rangle \quad (15)$$

where u represents the index of each agent, and $\hat{s}$ stores each agent’s global statuses and local observations. The term a is a representation of all agents’ activity, and each agent’s reward is part of the tensor r . The observation function is indicated by o , and p represents the likelihood of a transition from the current state to the following state.

More specifically, the game has N agents and strategies parameterized by $\theta = \{\theta_1, \dots, \theta_N\}$. The term $\pi = \{\pi_1, \dots, \pi_N\}$ represents the collection of all agent policies. For agent i , the gradient of the expected return, denoted as $J(\theta_i) = \mathbb{E}[R_i]$, may thus be expressed as follows:

$$\nabla_{\theta_i} J(\theta_i) = \mathbb{E}_{s \sim p^\mu, a_i \sim \pi_i} [\nabla_{\theta_i} \log \pi_i(a_i | o_i) Q_i^\pi(\mathbf{x}, a_1, \dots, a_N)] \quad (16)$$

In our setting, the total actions $a_1, \dots, a_N$ are fed into $Q_i^\pi(\mathbf{x}, a_1, \dots, a_N)$, which is a centralized action–value function that produces the Q-value for agent i along with some state data. In the simplest scenario, states might be the sum of the observations made by each agent, $(o_1, \dots, o_N)$, but if accessible, we could also incorporate additional state data. Agents are allowed to have any incentive systems, even ones that provide rival rewards in a hostile environment. However, in this paper, we set the reward function as the total of rewards for all agents since our research focuses on situations where all agents cooperate to achieve a common goal, resulting in cooperative rewards.

The mentioned concept can be expanded to apply to deterministic policies. Now that we have N continuous policies μ_{θ_i} parameterized by θ_i , we can express the gradient as follows:

$$\nabla_{\theta_i} J(\mu_i) = \mathbb{E}_{\mathbf{x}, a \sim \mathcal{D}} [\nabla_{\theta_i} \mu_i(a_i | o_i) \nabla_{a_i} Q_i^\mu(\mathbf{x}, a_1, \dots, a_N) |_{a_i = \mu_i(o_i)}] \quad (17)$$

where the transitions $(\mathbf{x}, \mathbf{x}', a_1, \dots, a_N, r_1, \dots, r_N)$ are stored in replay buffer $\mathcal{D}$, which stores all agent experiences.

The policies of other agents must be updated for Equation (17) to be applied. Knowing the observations and policies of other agents is not a particularly restricting assumption, as this information is typically available to all actors if our goal is to educate agents to exhibit sophisticated communicative behavior in simulation.

3.4. Genetic Algorithm

GA is a computational model that is inspired by Darwin’s biological evolution theory and is used for searching for optimal solutions by simulating natural evolution. It operates directly on structural objects, avoiding differentiation and function continuity constraints [34–36]. With inherent implicit parallelism and strong global optimization ability, it employs probabilistic optimization methods for automatically obtaining and guiding the search space without strict rules, allowing adaptive adjustments of the search direction. GA targets all individuals in a population and efficiently explores an encoded parameter space using randomization techniques. Its genetic operations include selection, crossover, and mutation. The core components of a GA are parameter encoding, initial population setting, fitness function design, genetic operation design, and control parameter setting. To demonstrate the operation of a GA, we consider an unconstrained optimization problem. The objective is to maximize the following function:

$$\text{Maximize } f(k), \quad k_n^l \leq k_n \leq k_n^u, \quad n = 1, 2, \dots, N. \quad (18)$$

The variable k_i can take values within the range of k_n^l and k_n^u . Although we consider a maximization problem, a GA can also be used for minimization problems. To ensure the proper functioning of the GA, the following steps are taken.

Variables k_i in Equation (18) are initially coded in specific string structures before using GAs to address the aforementioned issue. It is essential to mention that coding the variables is not always required at this stage. In some studies, GAs are directly applied to the variables, but for the sake of discussing the fundamental ideas of a simple GA, we will disregard these exceptions.

The fitness function is evaluated for each individual in the initial population and subsequently for each new generation after applying the genetic operators of selection, crossover, and mutation. Since each individual's fitness is independent of that of the others, parallel computation is feasible.

Such transitions can take many different forms. Below are two commonly used fitness mappings.

$$\mathcal{F}(k) = \frac{1}{1 + f(k)} \quad (19)$$

This transformation converts a minimization problem into an equivalent maximization problem without changing the position of the minimum. The objective function can be transformed using a different function to provide the fitness value $\mathcal{F}(i)$, as shown below:

$$\mathcal{F}(i) = V - \frac{O(i)P}{\sum_{i=1}^P O(i)} \quad (20)$$

where V is a large value to ensure non-negative fitness values, P is the population size, and $O(i)$ is the objective function value of the n th individual. For this study, V is chosen as the maximum value of the second term in Equation (20), leading to a fitness value of zero, which equals the maximum value of the objective function. This transformation does not alter the solution's position; it merely converts a minimization problem into an equivalent maximization problem. The term "string fitness" refers to the fitness function value of a string.

Genetic operators like selection, crossover, and mutation are applied to the population, producing a new generation based on the fitter individuals from the current generation. The selection operation picks individuals with advantages in the current population. The crossover or recombination operation creates descendants by exchanging a portion of chromosomes between two selected individuals, resulting in two new chromosomes representing offspring. The mutation operation randomly changes one or more chromosome values (genes) of each newly created individual. Mutations typically occur with a very low probability.

3.5. GA-MADDPG for Addressing Communication Coverage and Navigation in Its Own Abstract Formulation

In the abstract formulation in Section 3.1, the policy of the objective function can be expressed as $\pi(s_t) = a_t(s_t)$. In each episode j , the objective is to optimize the objective function by selecting the best coordination and optimal action (a) for each state (s). Different agents are assigned to navigate themselves to reach the target point, and each agent adopts an independent strategy. To address limitations and explore various scenarios, we use off-policy methods instead of on-policy methods since off-policy is more powerful and generalized. It ensures that the data are comprehensive and that all actions are covered. It can even come from a variety of sources—self-generated or external [37]. Figure 3 illustrates the highlights of the proposed GA-MADDPG.

All criticisms will be updated simultaneously to reduce the combined regression loss function for episode j :

$$\mathcal{L}(\theta_i) = \frac{1}{S} \sum_j \left(y^j - Q_i^\mu(\mathbf{x}^j, a_1^j, \dots, a_N^j) \right)^2 \quad (21)$$

The actor is updated using the sampled policy gradient:

$$\begin{aligned} \nabla_{\theta_i} J &\approx \frac{1}{S} \sum_j \nabla_{\theta_i} \mu_i(o_i^j) \\ &\quad \nabla_{a_i} Q_i^\mu(\mathbf{x}^j, a_1^j, \dots, a_i, \dots, a_N^j) \big|_{a_i = \mu_i(o_i^j)} \end{aligned} \quad (22)$$

And the centralized action–value function Q_i^μ is updated as:

$$\begin{aligned}\mathcal{L}(\theta_i) &= \mathbb{E}_{\mathbf{x}, a, r, \mathbf{x}'} [(Q_i^\mu(\mathbf{x}, a_1, \dots, a_N) - y)^2], \\ y &= r_i + \gamma Q_i^{\mu'}(\mathbf{x}', a'_1, \dots, a'_N) \big|_{a'_j = \mu'_j(o_j)}\end{aligned}\quad (23)$$

where

$$\mu' = \{\mu_{\theta'_1}, \dots, \mu_{\theta'_N}\} \quad (24)$$

is the collection of goal policies with postponed parameters θ_i .

The training process of the GA-MADPPG algorithm is summarized in Algorithm 1. We use off-policy DDPG training to maximize the reward.

Algorithm 1 GA-MADDPG algorithm

Require: Input state s , discount factor γ , and action a

Initialization : Initialize MPE environment with four agents (including 3 UGVs and 1 mobile BS); Initialize hyperparameter population.

$E_{\text{count}} = 0$

for $Episode = 1$ to max episode **do**

Reset environments, collect initial observations o_i of agents

for $step = 1$ to max step **do**

Choose A_t for each agent i

Agents take A_t and receive next observations o'_i

Calculate the total reward in Equation (6)

Store all agents' transitions in $\mathcal{D}$, and store the $\mathcal{L}$ of transitions in $\mathcal{D}$.

$E_{\text{count}} = E_{\text{count}} + 1$

if $E_{\text{count}} \geq \text{update episode}$ **then**

for $g = 1$ to critic updates steps **do**

Sample batch $\mathcal{B}$ from $\mathcal{D}$

Set $y^j = r_i^j + \gamma Q_i^{\mu'}(\mathbf{x}^j, a'_1, \dots, a'_N) \big|_{a'_k = \mu'_k(o_k^j)}$

Minimize the loss in Equation (21) to update critic

Update actor using the sampled policy gradient according to Equation (22)

Evaluate fitness of hyperparameter population according to Equation (19)

Crossover hyperparameter population

Mutation operation

Set new hyperparameter population according to $\mathcal{D}$.

end for

Update target parameters:

$\theta'_i \leftarrow \tau \theta_i + (1 - \tau) \theta'_i$

$E_{\text{count}} = 0$

end if

end for

end for

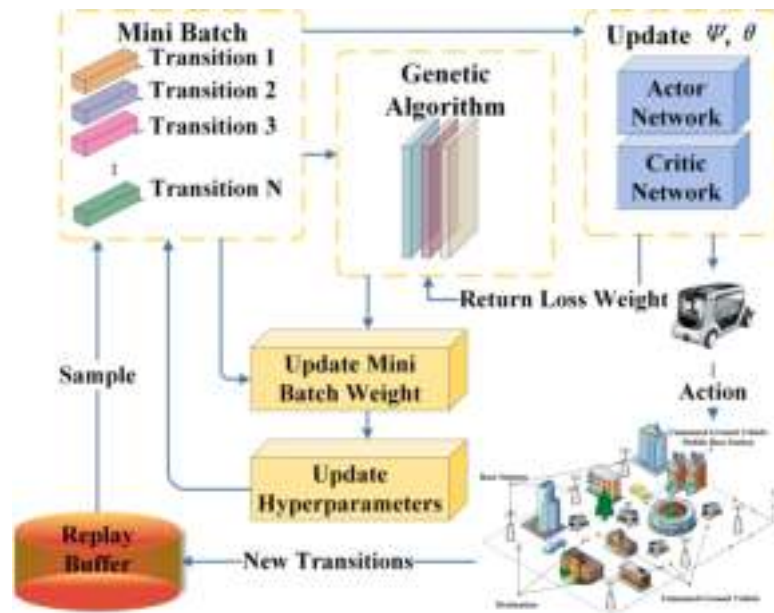


Figure 3. Detailed diagram of the GA-MADDPG algorithm.

4. Simulation Results

In this section, we present illustrative examples to depict the experimental setup of this paper. Based on these examples, we propose several metrics to assess the effectiveness of the algorithm and perform a quantitative analysis to clarify the advantages of our represented modeling approach and policy. Subsequently, we present numerical simulation results to showcase the effectiveness and efficiency of the algorithms. Additionally, we provide insightful comments on the results.

4.1. Settings of the Experiments

In this subsection, we present the precise experimental coefficient settings. The simulated area is a dense urban region of $2 \times 2 \text{ km}^2$ with seven cellular BS sites. In Figure 4, a top view of the channel model in this paper is shown, where seven ground base stations are represented by blue five-pointed stars, and the blue five-pointed star in the middle represents the movable base station. Each base station has three unit groups. Since there are seven base stations in total, the number of units is 21. The transmission power of the unit cell is set to $P_m = 20 \text{ dBm}$, the communication interruption threshold is set to $\gamma_{th} = 0 \text{ dB}$, and the noise power is defined as $\sigma^2 = -65 \text{ dBm}$. This paper adopts the base station antenna model required by the 3GPP specification. For simplicity, we assume that the UGVs' operational height is set at 0 m, disregarding the influence of terrain ups and downs. The specific values of the parameters involved in the simulated environment are as follows: the number of UGVs is set to four (including one movable BS), the number of obstacles is set to five in the main areas, and there are three target points. The positions of these elements are randomized each time they appear. As we employ a dynamic update mechanism for hyperparameters, we list the common parameters of the baseline algorithm and the GA-MADDPG algorithm in Table 1, and we also list the initial hyperparameter population of the GA-MADDPG algorithm in Table 2.

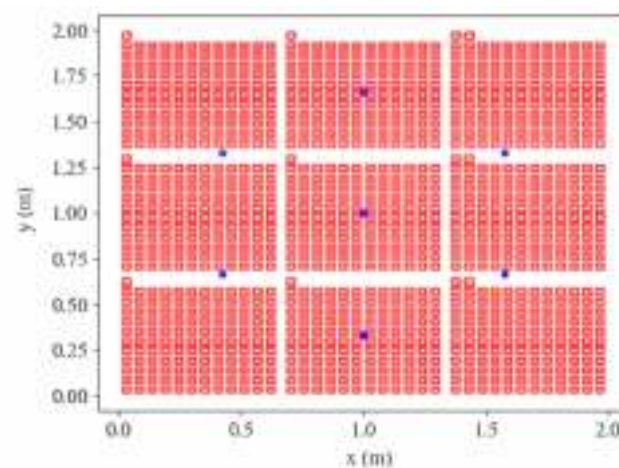
In this study, it is important to note that the communication environment is solely determined by the positioning of each UGV. The quality of communication among multiple UGVs does not influence their collaborative navigation. This is because the collaborative navigation process relies exclusively on a multi-agent algorithm to coordinate the UGVs in environmental exploration.

Table 1. GA-MADDPG parameter settings.

Definition	Value	Definition	Value
Max episodes	60,000	Minibatch size	512
Replay buffer capacity	1,000,000	Discount factor	0.99
Steps per update	100	Learning rate	0.0001
Max steps per episode	25	Update population rate	100
Time step length	1	Hidden dimension	64

Table 2. Initial hyperparameter population of GA-MADDPG algorithm.

Discount Factor	Learning Rate	Replay Buffer Capacity	Minibatch Size
0.9	0.01	10,000	512
0.95	0.001	100,000	1024
0.99	0.0005	1,000,000	2048

**Figure 4.** Plan view of base station model distribution.

4.2. Indicators of Evaluation for UGV Navigation

To objectively measure the navigational safety, effectiveness, robustness, and communication connection of UGVs, we have developed specific assessment indicators, which are detailed below. We also recorded the changing state of the evaluation metrics, as shown in Figure 5.

- **Communication return.** The communication return is the average communication quality per episode for the UGVs and is calculated based on Equation (1). The communication returns converge quickly from the initial -800 to -300 as shown by Figure 5a, which indicates that the communication quality has been improved and has stabilized in an interval.
- **Collision times:** The collision times are the sum of collisions between UGVs and obstacles and between drones and drones in an average round. The collision indicator converges from 540 to below 480, as shown by Figure 5b, indicating that the number of collisions has also been reduced somewhat, and since this study allows UGVs to have a certain number of collisions, the collision indicator is not the main optimization objective.
- **Outside times:** The outside times are the number of times the UGVs go out of bounds and run out of the environment we set. From Figure 5c, the rapid reduction in the number of times going out of bounds indicates that our research has significantly limited ineffective boundary violations, demonstrating that our study effectively operates within the designated area.

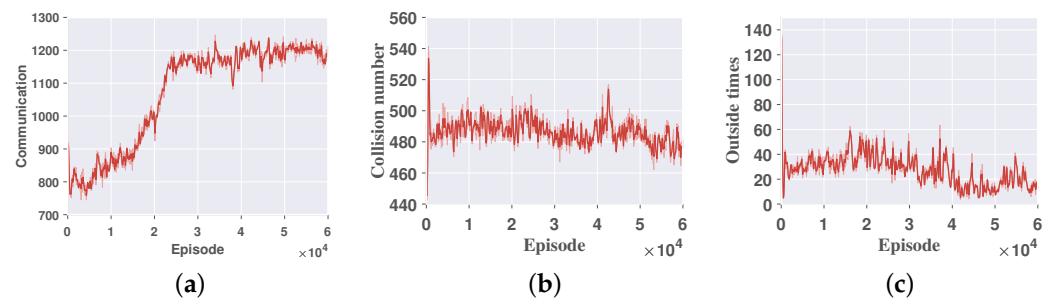


Figure 5. Three evaluation indicators for UGV navigation. (a) Communication return. (b) Collision number. (c) Outside times.

4.3. Comparative GA-MADDPG Experimentation

To compare with the suggested algorithm and determine whether the algorithm works better, we provide seven RL approaches that are thought of as baselines. The methods are MADDPG [24]: a classic multi-agent deep deterministic policy gradient, R-MADDPG [38]: a deep recurrent multi-agent actor–critic, MAPPO [39]: multi-agent proximal policy optimization, RMAPPO [39]: a deep recurrent multi-agent proximal policy optimization, MQMIX [40]: mellow–max monotonic value function factorization for deep multi-agent, MASAC [41]: a classic multi-agent soft actor–critic, MAD3PG [42]: a multi-agent deep distributional deterministic policy gradient, MATD3 [43]: the twin delayed deep deterministic policy gradient, and RMAPD3 [44]: the twin delayed deep deterministic policy gradient with a deep recurrent. Notably, we replicate these baselines using the same simulation environment to guarantee the experiment is fair.

The cumulative return of the GA-MADDPG and other algorithms, which is displayed in Figure 6, indicates the experimental comparison findings and highlights the potency of GA-MADDPG algorithms. GA-MADDPG outperforms the other algorithms by achieving a considerably higher reward return of about -1200 with 60,000 episodes, reaching its convergence point. Furthermore, as shown in Figure 6, both MADDPG and R-MADDPG achieve lower rewards of around -1600 compared to GA-MADDPG, providing strong evidence for the effectiveness of our contribution: the use of GA adaptive hyperparameters allows for better jumps out of the local optima and higher rewards. As shown in Figure 6, in the specific environment we configured, neither the original MADDPG algorithm nor its variant incorporating deep recurrent networks outperforms GA-MADDPG in areas of convergence speed and final convergence outcomes: GA-MADDPG converges in about 2000 episodes, while R-MADDPG converges in about 5000 episodes, and the original algorithm MADDPG converges even worse. Of greater significance, our experimental findings reveal that MASAC, MAPPO, MAD3PG, MQMIX, and RMAPPO encounter challenges in achieving a desirable convergence state within the multi-agent cooperative environment we constructed. MASAC required approximately 25,000 episodes to converge, ultimately stabilizing at a reward value of approximately -1800 . MAPPO and RMAPPO exhibited less stable convergence, with rewards fluctuating between -2000 and -2500 . Meanwhile, MAD3PG's reward converged to approximately -2100 . Regarding MQMIX, its reward demonstrated initial oscillation over the first 25,000 episodes, followed by a steady decline thereafter. This further emphasizes the superiority of GA-MADDPG in terms of performance and effectiveness.

Furthermore, certain algorithms tend to converge to local optima, which further reinforces the effectiveness of our decision to adopt the MADDPG algorithm and enhance it. As depicted in Figure 6, in the initial 25,000 episodes, GA-MADDPG may succumb to local optimality. However, the incorporation of the GA mechanism enables GA-MADDPG to attain elevated rewards beyond this threshold. Notably, MAPPO and MQMIX demonstrate subpar performance, possibly due to the lack of adaptive hyperparameter updates, hindering their effective cooperation within the multi-agent environment and leading

to convergence challenges. Therefore, this observation naturally demonstrates the high effectiveness of incorporating GA into multi-agent RL algorithms. By introducing GA, multi-agent algorithms can more effectively avoid falling into local optima, resulting in improved convergence speed and outcomes. And the variation of the loss calculated by Equation (21) is represented by Figure 7, from which we can see the constant convergence of the loss to near 1800, which can prove the convergence of the algorithm. During the validation process, Figure 8 displays several simulated paths of UGVs. Under optimal communication conditions, the BS UGV might remain stationary to prevent potential losses due to collisions. However, in situations with less than excellent communication, the BS UGV proactively moves to compensate for communication limitations. Additionally, statistics for the three evaluation indicators (Figure 5) show the improvement in communication return, the reduction in collision number, and the decrease in outside number as the algorithm converges. The return on communications exhibited an improvement from an initial value of -800 to -300 towards the conclusion of the experiment. Concurrently, the frequency of collisions decreased from 540 to 470, and the occurrences of external events diminished from 100 to nearly zero. This suggests that as the algorithm converges, the three evaluation metrics also reach optimality.

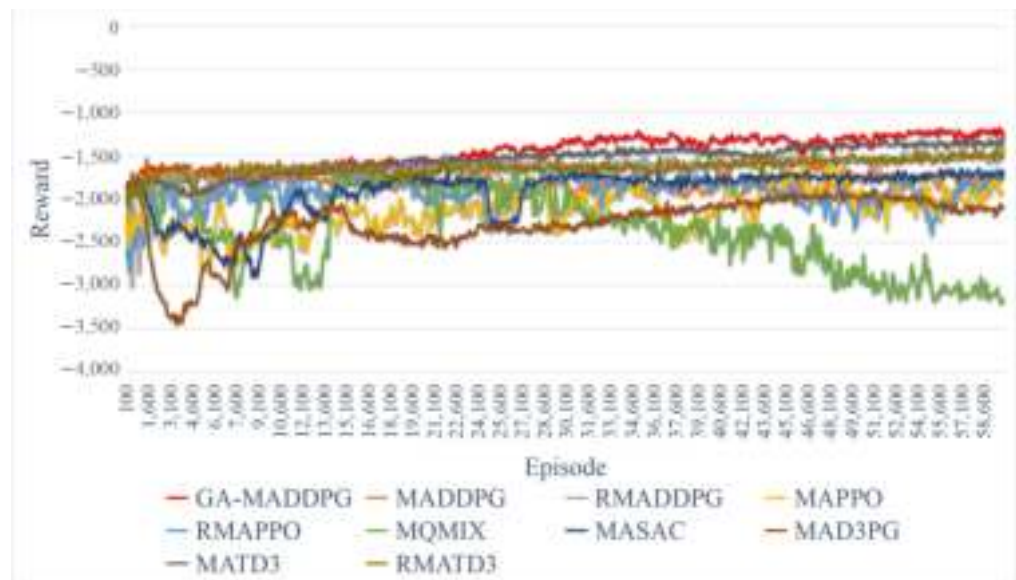


Figure 6. Average cost of the GA-MADDPG and other advanced algorithms.

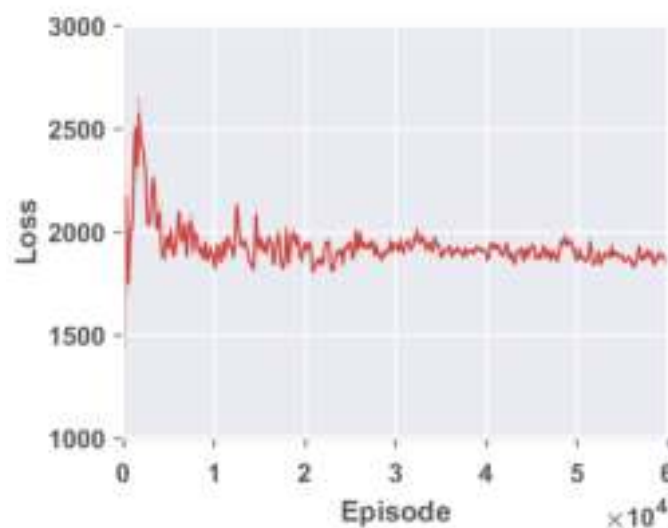


Figure 7. Evolution of loss function.

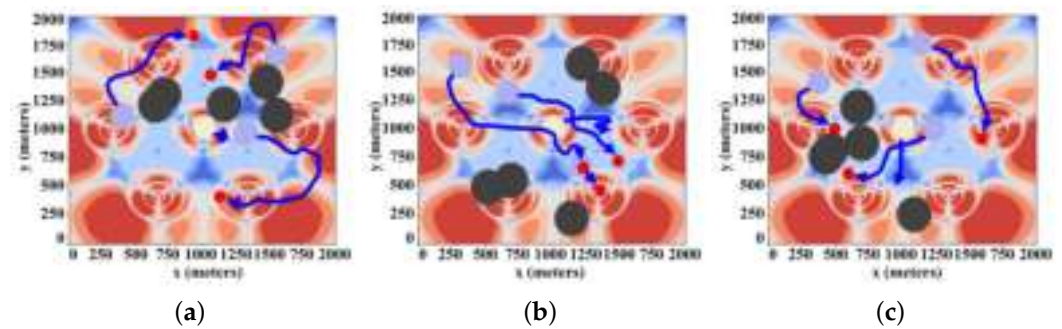


Figure 8. Some UGV path maps based on GA-MADDPG.

4.4. Generalization Experiment of GA-MADDPG

4.4.1. Simulation with Different Numbers of UGVs

To further prove the universality of the proposed GA-MADDPG algorithm in the set environment, this study also designed two other generalization experiments for the scene. The experiment set different numbers of UGVs, target points, and obstacles in the scene to determine whether the algorithm GA-MADDPG can continue to perform superiorly. It should be noted that since some baseline algorithms in Section 4.3 have performed poorly or even have difficulty converging, the generalization experiment uses four baseline algorithms that are relatively stable in Section 4.3, including MASAC, MAD3PG, MADDPG, and its variant, RMADDPG. Generalization environment 1: The number of UGVs increases to four, the number of mobile base stations is one, the number of target points increases to four, and the number of obstacles increases to seven. The significance of setting up the environment in this way is to increase the severity of the environment by increasing the number of UGVs and the number of obstacles.

From Figure 9, we can see that despite the increased complexity of the environment, the GA-MADDPG algorithm always has a higher convergence value in harsh environments and can converge to a high value well. The GA-MADDPG algorithm can maintain convergence to a reward value of -3000 , while the other baseline algorithms do not perform well or even find it difficult to converge in complex environments, and the highest reward value is only around -3300 . This fully demonstrates that the GA-MADDPG algorithm still has better performance than other algorithms after the environmental complexity increases.

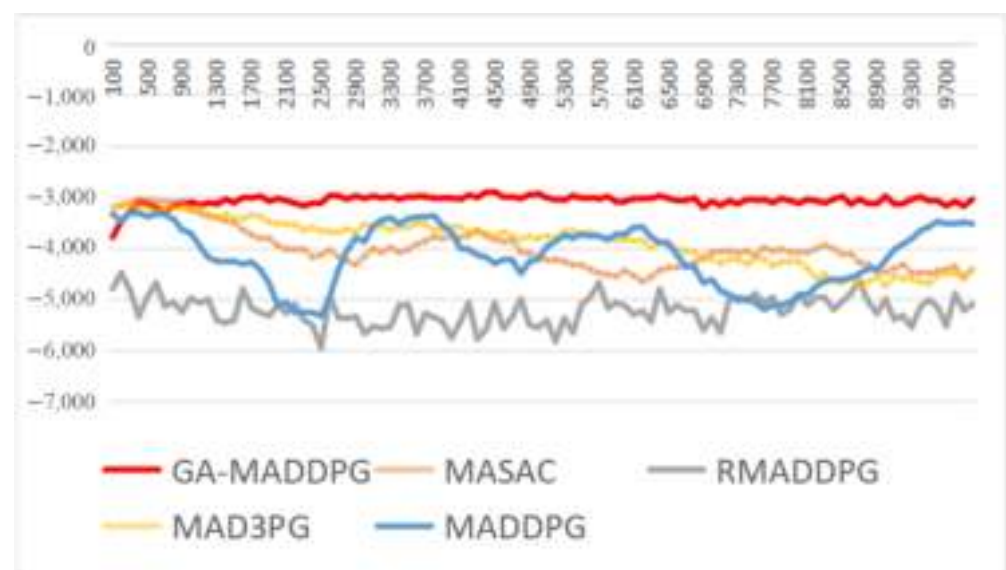


Figure 9. Average cost comparison of generalization environment 1.

Generalized environment 2: The number of UGVs is reduced to two, the number of mobile base stations is one, the number of target points is reduced to two, and the number of obstacles is reduced to three. The significance of setting up the environment in this way is to improve the simplicity of the environment by simplifying the number of UGVs and obstacles so that the UGV can complete the goal with a greater reward.

As can be seen from Figure 10, the rewards of most algorithms show a good upward trend. This is because the generalized environment uses a simpler three UGVs (including a UGV base station), three obstacles, and two target points. The algorithm performs better in a simple environment and convergence is easier than for the generalized environment. As the number of vehicles decreases, the number of collisions and out-of-bounds also decrease accordingly. It should be noted that since the communication environment parameters remain unchanged, the reward value of the overall algorithm is positive, which is normal. From Figure 10, it can be seen that in this generalized environment, the reward of the GA-MADDPG algorithm always remains ahead, both in terms of convergence speed and final convergence value, which are much higher than for the other algorithms, and the final reward value can converge to about 200. As a basic algorithm, MADDPG also has a higher convergence value of about 150. This fully demonstrates that the GA-MADDPG algorithm can also perform well in a simple environment.



Figure 10. Average cost comparison of generalization environment 2.

It can be seen from Figures 9 and 10 that in the experimental environments with two different parameter settings, despite changes in the number of UGVs, the number of target points, and the number of obstacles, the GA-MADDPG algorithm can still perform better than the other algorithms, which fully demonstrates the robustness of the GA-MADDPG algorithm and its universality to environmental scenarios.

4.4.2. Experiments on the Effectiveness of the Mobile BS

The previous subsections prove the stability and convergence of our proposed algorithm. Also, the last section proves that our proposed algorithm is superior in the same scenario. To better demonstrate the effectiveness of the mobile base station proposed in this paper, we add an extra experiment: only changing the mobile BS to a fixed BS but using the same algorithm.

We use the communication return as an evaluation metric, and the communication return with a mobile base station is better than that of the fixed base station from the beginning of training, as shown by Figure 11. The communication return of a single UGV

can eventually converge to around 300, while that of the fixed base station hovers around 200 feet, which fully proves the effectiveness of our proposed mobile base station.

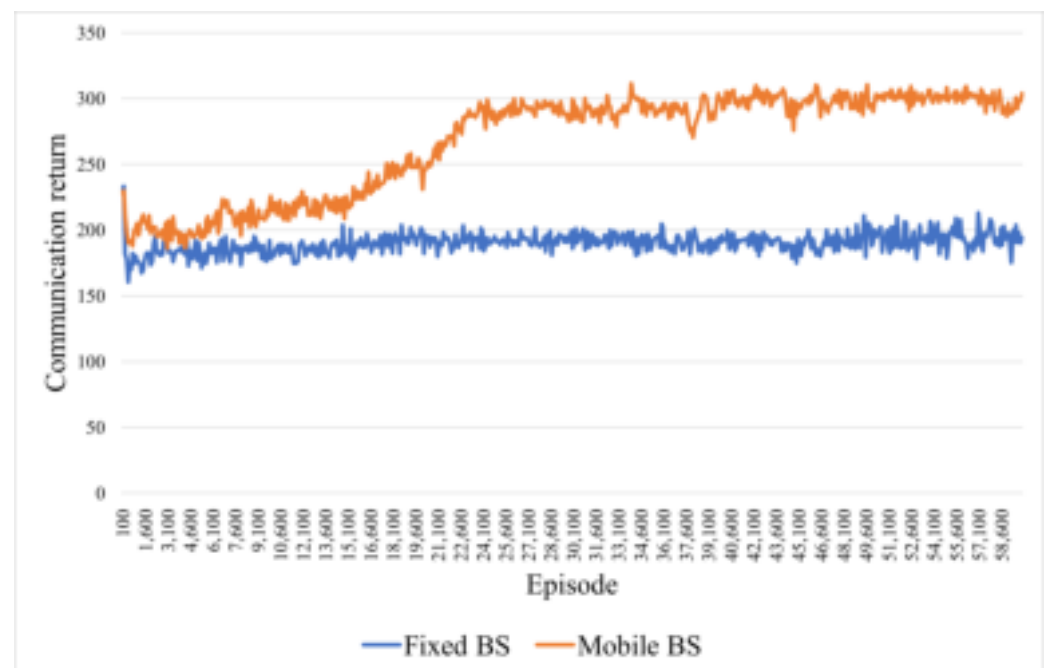


Figure 11. Comparison of communication returns between mobile BS and fixed BS.

5. Conclusions

In this article, a cooperative system for multi-UGV cooperative navigation within a communication coverage area is proposed. The system is formulated as an MDP to determine an optimal navigation policy for the UGVs, with the aim of maximizing the total reward. In contrast to prior studies focusing on fixed coverage-aware navigation, this paper introduces a novel approach by incorporating a mobile BS into the multi-intelligent-body algorithm. This innovation aims to enhance communication coverage and expand the solution space available for intelligent agents. To mitigate the risk of local optima, this study introduces a GA hyperparameter adaptive updating mechanism to address the multi-UGV navigation problem. We coin the term GA-MADDPG to refer to this novel RL algorithm. The simulation results demonstrate that GA-MADDPG exhibits favorable performance, convergence rates, and effectiveness compared to other RL algorithms.

In our future research, we would like to address the following points: (1) To enhance model realism, one can combine a traditional PID control with multi-agent RL and further optimize the navigation policy by taking control of the machine operation. (2) One can try to use a new architecture to learn policies, such as by using LSTM (long short-term memory) and the transformer architecture. LSTM can solve the problem of gradient vanishing and gradient explosion during the training of long sequences; the advantage of the transformer architecture is that its attention layer can learn a sequence of actions very well.

Author Contributions: Research design, X.L. and M.H.; data acquisition, X.L.; writing—original draft preparation, X.L.; writing—review and editing, X.L. and M.H.; supervision, M.H.; funding acquisition, M.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Academician Innovation Platform Special Project of Hainan Province (YSPTZX202209).

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Afzali, S.R.; Shoaran, M.; Karimian, G. A Modified Convergence DDPG Algorithm for Robotic Manipulation. *Neural Process. Lett.* **2023**, *55*, 11637–11652. [\[CrossRef\]](#)
2. Chai, R.; Niu, H.; Carrasco, J.; Arvin, F.; Yin, H.; Lennox, B. Design and experimental validation of deep reinforcement learning-based fast trajectory planning and control for mobile robot in unknown environment. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *35*, 5778–5792. [\[CrossRef\]](#)
3. Dong, X.; Wang, Q.; Yu, J.; Lü, J.; Ren, Z. Neuroadaptive Output Formation Tracking for Heterogeneous Nonlinear Multiagent Systems with Multiple Nonidentical Leaders. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *35*, 3702–3712. [\[CrossRef\]](#)
4. Wang, Y.; Zhao, C.; Liang, J.; Wen, M.; Yue, Y.; Wang, D. Integrated Localization and Planning for Cruise Control of UGV Platoons in Infrastructure-Free Environments. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 10804–10817. [\[CrossRef\]](#)
5. Tran, V.P.; Perera, A.; Garratt, M.A.; Kasmarik, K.; Anavatti, S.G. Coverage Path Planning with Budget Constraints for Multiple Unmanned Ground Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 12506–12522. [\[CrossRef\]](#)
6. Wu, Y.; Li, Y.; Li, W.; Li, H.; Lu, R. Robust Lidar-Based Localization Scheme for Unmanned Ground Vehicle via Multisensor Fusion. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 5633–5643. [\[CrossRef\]](#)
7. Zhang, W.; Zuo, Z.; Wang, Y. Networked multiagent systems: Antagonistic interaction, constraint, and its application. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 3690–3699. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Chen, D.; Weng, J.; Huang, F.; Zhou, J.; Mao, Y.; Liu, X. Heuristic Monte Carlo algorithm for unmanned ground vehicles realtime localization and mapping. *IEEE Trans. Veh. Technol.* **2020**, *69*, 10642–10655. [\[CrossRef\]](#)
9. Unlu, H.U.; Patel, N.; Krishnamurthy, P.; Khorrami, F. Sliding-window temporal attention based deep learning system for robust sensor modality fusion for UGV navigation. *IEEE Robot. Autom. Lett.* **2019**, *4*, 4216–4223. [\[CrossRef\]](#)
10. Lyu, X.; Hu, B.; Wang, Z.; Gao, D.; Li, K.; Chang, L. A SINS/GNSS/VDM integrated navigation fault-tolerant mechanism based on adaptive information sharing factor. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–13. [\[CrossRef\]](#)
11. Sun, C.; Ye, M.; Hu, G. Distributed optimization for two types of heterogeneous multiagent systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 1314–1324. [\[CrossRef\]](#)
12. Shan, Y.; Fu, Y.; Chen, X.; Lin, H.; Lin, J.; Huang, K. LiDAR based Traversable Regions Identification Method for Off-road UGV Driving. *IEEE Trans. Intell. Veh.* **2023**, *9*, 3544–3557. [\[CrossRef\]](#)
13. Garaffa, L.C.; Basso, M.; Konzen, A.A.; de Freitas, E.P. Reinforcement learning for mobile robotics exploration: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *34*, 3796–3810. [\[CrossRef\]](#)
14. Huang, C.Q.; Jiang, F.; Huang, Q.H.; Wang, X.Z.; Han, Z.M.; Huang, W.Y. Dual-graph attention convolution network for 3-D point cloud classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *35*, 4813–4825. [\[CrossRef\]](#)
15. Nguyen, H.T.; Garratt, M.; Bui, L.T.; Abbass, H. Supervised deep actor network for imitation learning in a ground-air UAV-UGVs coordination task. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–8.
16. Han, Z.; Yang, Y.; Wang, W.; Zhou, L.; Gadekallu, T.R.; Alazab, M.; Gope, P.; Su, C. RSSI Map-Based Trajectory Design for UGV Against Malicious Radio Source: A Reinforcement Learning Approach. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 4641–4650. [\[CrossRef\]](#)
17. Feng, Z.; Huang, M.; Wu, Y.; Wu, D.; Cao, J.; Korovin, I.; Gorbachev, S.; Gorbacheva, N. Approximating Nash equilibrium for anti-UAV jamming Markov game using a novel event-triggered multi-agent reinforcement learning. *Neural Netw.* **2023**, *161*, 330–342. [\[CrossRef\]](#)
18. Huang, X.; Deng, H.; Zhang, W.; Song, R.; Li, Y. Towards multi-modal perception-based navigation: A deep reinforcement learning method. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4986–4993. [\[CrossRef\]](#)
19. Wu, S.; Xu, W.; Wang, F.; Li, G.; Pan, M. Distributed federated deep reinforcement learning based trajectory optimization for air-ground cooperative emergency networks. *IEEE Trans. Veh. Technol.* **2022**, *71*, 9107–9112. [\[CrossRef\]](#)
20. Watkins, C.J.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [\[CrossRef\]](#)
21. Tran, T.H.; Nguyen, M.T.; Kwok, N.M.; Ha, Q.P.; Fang, G. Sliding mode-PID approach for robust low-level control of a UGV. In Proceedings of the 2006 IEEE International Conference on Automation Science and Engineering, Shanghai, China, 8–10 October 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 672–677.
22. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized experience replay. *arXiv* **2015**, arXiv:1511.05952.
23. Sutton, R.S.; Barto, A.G. Reinforcement Learning: An Introduction. *IEEE Trans. Neural Netw.* **1998**, *9*, 1054. [\[CrossRef\]](#)
24. Lowe, R.; Wu, Y.I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv* **2017**, arXiv:1706.02275.
25. Mirjalili, S.; Mirjalili, S. Genetic algorithm. *Evolutionary Algorithms and Neural Networks: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 43–55.
26. Sehgal, A.; La, H.; Louis, S.; Nguyen, H. Deep reinforcement learning using genetic algorithm for parameter optimization. In Proceedings of the 2019 Third IEEE International Conference on Robotic Computing (IRC), Naples, Italy, 25–27 February 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 596–601.
27. Chen, R.; Yang, B.; Li, S.; Wang, S. A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem. *Comput. Ind. Eng.* **2020**, *149*, 106778. [\[CrossRef\]](#)

28. Alipour, M.M.; Razavi, S.N.; Feizi Derakhshi, M.R.; Balafar, M.A. A hybrid algorithm using a genetic algorithm and multiagent reinforcement learning heuristic to solve the traveling salesman problem. *Neural Comput. Appl.* **2018**, *30*, 2935–2951. [\[CrossRef\]](#)
29. Liu, Z.; Chen, B.; Zhou, H.; Koushik, G.; Hebert, M.; Zhao, D. Mapper: Multi-agent path planning with evolutionary reinforcement learning in mixed dynamic environments. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 11748–11754.
30. Huang, M.; Lin, X.; Feng, Z.; Wu, D.; Shi, Z. A multi-agent decision approach for optimal energy allocation in microgrid system. *Electr. Power Syst. Res.* **2023**, *221*, 109399. [\[CrossRef\]](#)
31. Qiu, C.; Hu, Y.; Chen, Y.; Zeng, B. Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications. *IEEE Internet Things J.* **2019**, *6*, 8577–8588. [\[CrossRef\]](#)
32. Littman, M.L. Markov games as framework for multi-agent reinforcement learning. In Proceedings of the Proc International Conference on Machine Learning, New Brunswick, NJ, USA, 10–13 July 1994; pp. 157–163.
33. Feng, Z.; Huang, M.; Wu, D.; Wu, E.Q.; Yuen, C. Multi-Agent Reinforcement Learning with Policy Clipping and Average Evaluation for UAV-Assisted Communication Markov Game. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 14281–14293. [\[CrossRef\]](#)
34. Liu, H.; Zong, Z.; Li, Y.; Jin, D. NeuroCrossover: An intelligent genetic locus selection scheme for genetic algorithm using reinforcement learning. *Appl. Soft Comput.* **2023**, *146*, 110680. [\[CrossRef\]](#)
35. Köksal Ahmed, E.; Li, Z.; Veeravalli, B.; Ren, S. Reinforcement learning-enabled genetic algorithm for school bus scheduling. *J. Intell. Transp. Syst.* **2022**, *26*, 269–283. [\[CrossRef\]](#)
36. Chen, Q.; Huang, M.; Xu, Q.; Wang, H.; Wang, J. Reinforcement Learning-Based Genetic Algorithm in Optimizing Multidimensional Data Discretization Scheme. *Math. Probl. Eng.* **2020**, *2020*, 1698323. [\[CrossRef\]](#)
37. Yang, J.; Sun, Z.; Hu, W.; Steinmeister, L. Joint control of manufacturing and onsite microgrid system via novel neural-network integrated reinforcement learning algorithms. *Appl. Energy* **2022**, *315*, 118982. [\[CrossRef\]](#)
38. Shi, H.; Liu, G.; Zhang, K.; Zhou, Z.; Wang, J. MARL Sim2real Transfer: Merging Physical Reality with Digital Virtuality in Metaverse. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 2107–2117. [\[CrossRef\]](#)
39. Yu, C.; Velu, A.; Vinitzky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The surprising effectiveness of ppo in cooperative multi-agent games. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24611–24624.
40. Rashid, T.; Farquhar, G.; Peng, B.; Whiteson, S. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 10199–10210.
41. Wu, T.; Wang, J.; Lu, X.; Du, Y. AC/DC hybrid distribution network reconfiguration with microgrid formation using multi-agent soft actor-critic. *Appl. Energy* **2022**, *307*, 118189. [\[CrossRef\]](#)
42. Yan, C.; Xiang, X.; Wang, C.; Li, F.; Wang, X.; Xu, X.; Shen, L. PASCAL: PopulAtion-Specific Curriculum-based MADRL for collision-free flocking with large-scale fixed-wing UAV swarms. *Aerosp. Sci. Technol.* **2023**, *133*, 108091. [\[CrossRef\]](#)
43. Ackermann, J.J.; Gabler, V.; Osa, T.; Sugiyama, M. Reducing Overestimation Bias in Multi-Agent Domains Using Double Centralized Critics. *arXiv* **2019**, arXiv:1910.01465.
44. Xing, X.; Zhou, Z.; Li, Y.; Xiao, B.; Xun, Y. Multi-UAV Adaptive Cooperative Formation Trajectory Planning Based on an Improved MATD3 Algorithm of Deep Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2024**. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Hybrid Machine Learning for Automated Road Safety Inspection of Auckland Harbour Bridge

Munish Rathee ^{1,*}, Boris Bačić ^{1,*} and Maryam Doborjeh ^{1,2}

¹ School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand; mgholami@aut.ac.nz or maryam.gholami.doborjeh@aut.ac.nz

² Knowledge Engineering and Discovery Research Innovation, Auckland University of Technology, Auckland 1010, New Zealand

* Correspondence: munishonthenet@gmail.com or munish.rathee@aut.ac.nz (M.R.); boris.bacic@aut.ac.nz (B.B.)

Abstract: The Auckland Harbour Bridge (AHB) utilises a movable concrete barrier (MCB) to regulate the uneven bidirectional flow of daily traffic. In addition to the risk of human error during regular visual inspections, staff members inspecting the MCB work in diverse weather and light conditions, exerting themselves in ergonomically unhealthy inspection postures with the added weight of protection gear to mitigate risks, e.g., flying debris. To augment visual inspections of an MCB using computer vision technology, this study introduces a hybrid deep learning solution that combines kernel manipulation with custom transfer learning strategies. The video data recordings were captured in diverse light and weather conditions (under the safety supervision of industry experts) involving a high-speed (120 fps) camera system attached to an MCB transfer vehicle. Before identifying a safety hazard, e.g., the unsafe position of a pin connecting two 750 kg concrete segments of the MCB, a multi-stage preprocessing of the spatiotemporal region of interest (ROI) involves a rolling window before identifying the video frames containing diagnostic information. This study utilises the ResNet-50 architecture, enhanced with 3D convolutions, within the STENet framework to capture and analyse spatiotemporal data, facilitating real-time surveillance of the Auckland Harbour Bridge (AHB). Considering the sparse nature of safety anomalies, the initial peer-reviewed binary classification results (82.6%) for safe and unsafe (intervention-required) scenarios were improved to 93.6% by incorporating synthetic data, expert feedback, and retraining the model. This adaptation allowed for the optimised detection of false positives and false negatives. In the future, we aim to extend anomaly detection methods to various infrastructure inspections, enhancing urban resilience, transport efficiency and safety.

Keywords: anomaly detection; structural damage detection; traffic safety; computer vision; machine learning; deep learning; transfer learning; ARDAD



Citation: Rathee, M.; Bačić, B.; Doborjeh, M. Hybrid Machine Learning for Automated Road Safety Inspection of Auckland Harbour Bridge. *Electronics* **2024**, *13*, 3030. <https://doi.org/10.3390/electronics13153030>

Academic Editors: Jiwei Zhang and Shaozhang Niu

Received: 15 June 2024

Revised: 24 July 2024

Accepted: 29 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Auckland Harbour Bridge, spanning 1.2 km across Waitemata Harbour, was opened on 30 May 1959. Initially handling 11,205 vehicles daily, the bridge currently accommodates around 154,000 vehicles daily, with peaks over 200,000 due to public transport shifts [1]. The bridge supports rapid regional development, with a quadrupled North Shore population in the past 50 years. The New Zealand Transport Authority (NZTA), also known as Waka Kotahi, annually invests up to NZD 4 million in its maintenance and employs about 160 people for ongoing upgrades and maintenance. Like all crucial transportation infrastructure in New Zealand, the Auckland Harbour Bridge (AHB) faces significant maintenance challenges due to environmental and external factors. Such challenges contribute to the country's overall high maintenance costs for road infrastructure, which amount to as much as 1.1% of New Zealand's GDP [2].

This paper is organised as follows:

Section 1 presents an overview of the problem and the research question; Section 2 includes a comprehensive literature review on Automated Road Defect and Anomaly Detection; and Section 3 proposes a new methodology based on a hybrid deep learning solution to augment visual inspections using computer vision technology.

1.1. Background

Installed in 1990, the movable concrete barrier (MCB) on the AHB has enhanced rush-hour traffic flow and safety. Every weekday, the AHB utilises two Barrier Transfer Machines (BTMs) to adjust lane configurations, improving traffic flow during peak hours by moving 750 kg concrete blocks. The BTMs, essential for managing the daily tidal, were introduced in 2009 for 1.4 million NZD each. Despite the high cost, the BTMs do not have an automated surveillance system to ensure the integrity of the movable concrete barriers (MCBs). In the absence of an MCB barrier safety inspection system, NZTA staff must walk over a mile in hazardous conditions and amidst dangerous traffic to manually inspect and ensure the integrity of the MCB [3].

The reliability of the MCB system depends on the integrity of the metal pins connecting the barrier segments (Figure 1). The critical function of the pins is to secure barrier segments that regulate lane division based on traffic flow [3]. Malfunctioning or dislodged pins pose risks to traffic safety and impede the system's efficiency, leading to potential traffic disruptions and increased accident risks.

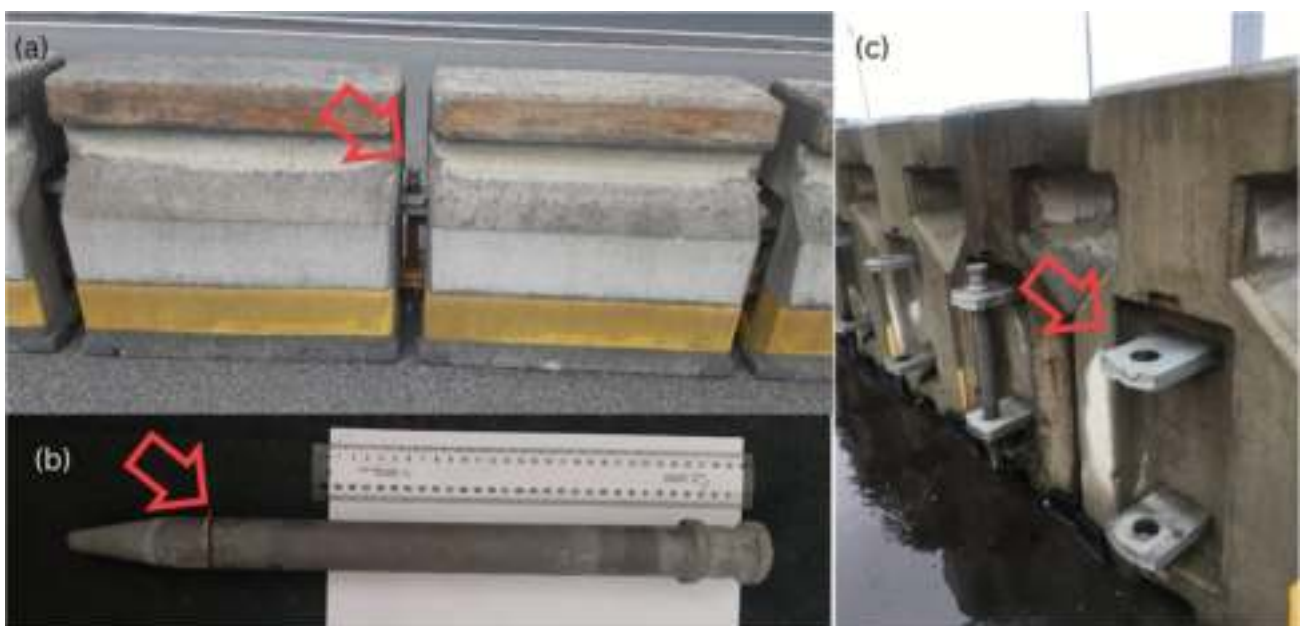


Figure 1. A movable concrete barrier system and its safety challenges. (a) Pin position requiring fixing, (b) metal pin with detachable safety ring, and (c) movable concrete barrier joints without metal pins.

Prior works in automated infrastructure inspection often fall short in dynamic and complex environments like the AHB [4–6]. This research aims to advance the field by incorporating a novel hybrid deep learning and spatiotemporal data analysis, allowing for a more accurate and reliable detection of safety anomalies in complex environments such as the AHB. This study employs spatiotemporal analysis, deployable AI algorithms, and semi-automated synthetic data generation to enhance traffic barrier monitoring, transforming research into practical, real-time anomaly detection solutions. The objective is to reduce the risks linked with manual inspections, enhance traffic safety (Figure 2), and boost the efficiency of barrier systems. Building on the Proof-of-Concept developed in 2019–2020 [7], this research utilises computer vision and deep learning techniques to au-

tomate the detection of metal pins in movable concrete barriers (MCBs) on the Auckland Harbour Bridge (AHB).



Figure 2. The scale of the challenges for manual inspection and fixing of the metal pin: (a) various locations of movable concrete barriers on and around the Auckland Harbour Bridge; (b) finding an unsafe pin and fixing it manually.

Hybrid machine-learning models, which integrate CNNs with other machine-learning techniques, can revolutionise inspections by enhancing the accuracy and reliability of detecting and classifying anomalous conditions [8,9]. The deep learning models trained on the synthetically enhanced dataset use a feature-based detection approach to analyse video frames for signs of pin displacement. Real-time monitoring capability is crucial for effective traffic management, especially during peak hours. Immediate alerts to bridge operators facilitate timely interventions, preventing potential safety issues from escalating into more severe incidents. Developing such an automated surveillance system advances civil engineering and traffic safety, offering a scalable, efficient solution to a longstanding safety challenge and setting a precedent for similar applications worldwide, potentially leading to the broader adoption of intelligent traffic management solutions in global urban settings [10].

This study developed a hybrid machine learning system for real-time, privacy-preserving anomaly detection in road safety inspections. This research's contributions are listed as follows:

- Produces a traffic safety analysis artefact scalable to scenarios in over 20+ countries and hundreds of similar traffic scenes [10] that employ movable concrete barriers.
- Introduces a semi-automated synthetic data generation method using a novel background cloning technique. The novel approach addresses data sparsity and enhances model training, with repeatability value for other computer vision case studies facing dataset balancing issues.
- Refines classification methods to balance false positives and negatives, improving detection accuracy from 82.6% (reported in earlier peer-reviewed research [11]) to 93.6%. Achieving this 11% increase in accuracy within complex traffic scenes characterised by chaotic backgrounds and lighting conditions underscores the artefact's viability for real-world applications.
- Successfully navigates hazardous traffic scenes for data collection by adhering to industry safety protocols. This repeatable approach provides a comprehensive blueprint for managing similar scenarios, ensuring stakeholder satisfaction and achieving sufficient data.

1.2. Research Questions and Modelling Concepts

This paper presents the development of automated solutions leveraging computer vision and deep learning to enhance traffic safety and operational efficiency by automating the inspection of metal pins in movable concrete barriers. This research explores visual sensors' potential to detect transport activity anomalies while maintaining privacy. Focused on the Auckland Harbour Bridge's movable barrier, our research automates safety screening and enhances anomaly detection for critical yet underrepresented classes like unsafe pin positions. Furthermore, classification techniques were refined to balance false positives and negatives, thereby improving the reliability and effectiveness of the traffic safety system.

1. Can relevant information from visual sensors be extracted for anomaly detection in transport activity while preserving privacy?
 - (a) What are the optimal methods for extracting vital information from visual media to detect transport activity anomalies without compromising privacy?
 - (b) To what degree can the safety screening of Auckland Harbour Bridge's movable barrier and the surrounding areas be automated using AI, CV, and DL methods?
2. How can synthetic data generation be streamlined to portray minority classes, such as unsafe pin positions, in anomaly detection tasks within traffic safety?
3. How can classification performance be honed to optimise an equilibrium between false positives and negatives from the early Proof-of-Concept (PoC) [7] while considering present and anticipated data scenarios?

2. Literature Review

Automated Road Defect and Anomaly Detection (ARDAD) uses computer vision, combining traditional and deep learning methods with unsupervised learning [12,13]. Traditional unsupervised methods, relying on datasets of normal conditions, often produce high false alarm rates in complex environments [14]. This research employs the Spatio-Temporal Enhanced Network (STENet) to address the problem of complex traffic scenes, which leverages temporal and spatial data for better generalisation and robust anomaly detection.

2.1. Auckland Harbour Bridge

The New Zealand Transport Agency (NZTA) manages the Auckland Harbour Bridge (AHB), an eight-lane motorway supporting over 200,000 vehicles daily. Installed in the 1990s, the movable concrete lane barrier (MCB) is essential for preventing crashes and optimising traffic flow during peak hours. The MCB system, consisting of 750 kg concrete blocks connected by metal pins, is prone to displacement due to traffic and ambient vibrations, posing significant risks [1,3,15]. Barrier Transfer Machines (BTMs) facilitate the movement of the MCB, typically adjusting lanes four times daily to manage traffic flow effectively.

Manual inspection of the pins is labour-intensive and occurs under poor ergonomic conditions, making it susceptible to human error. Despite the system's efficiency in altering traffic lanes swiftly—approximately 10 min for a one-kilometre section—the lack of automated pin inspection mechanisms within the existing NZD 1.4 million BTMs today may be seen as a significant oversight and an opportunity to apply CV in maintenance and safety protocols [1,15,16]. While effective in managing contraflow and heavy occupancy lanes, the movable concrete barrier (MCB) system requires continual manual monitoring to ensure its structural integrity and operational reliability.

2.2. Review of Surveys on ARDAD

The application of artificial intelligence (AI) and deep learning in road infrastructure anomaly and object detection has achieved significant breakthroughs over the past decade [5,12,13]. AI advancements have impacted domains such as autonomous driving,

face recognition, and personalised healthcare [17,18]. Deep learning techniques, particularly Convolutional Neural Networks (CNNs), have revolutionised image processing by automating segmentation, recognition, and reconstruction tasks [6]. CNNs efficiently process and classify image data through pooling and convolution, extracting features and reducing dimensionality [19–22]. Unlike traditional methods that rely on hand-engineered features, deep learning models learn feature representations directly from data. Architectures like R-CNN and its variants excel in object detection using region proposal methods to pre-selected areas of interest [23].

Moreover, transfer learning enhances model generalizability by applying models trained on one task to related but different tasks [24]. Transfer learning leverages pre-trained models to improve performance on new datasets, often by fine-tuning the final layers of networks like ResNet-50, which demonstrated remarkable success in the Visual Recognition Challenge [25]. The approach underscores deep learning models' versatility and adaptive capacity in handling diverse and complex visual data environments.

2.3. Video and Image Optimisation Techniques in ARDAD

Selecting an appropriate colour space in image segmentation is crucial and often application-specific, with no consensus on the best choice. Common colour spaces include RGB, LAB, CMY, XYZ, HSV, YCbCr, YIQ, YUV, and DHT [26,27]. For example, RGB is effective for bilirubin concentration changes (Equation (1)), while CMY and HSV excel in other tasks [28]. Advanced methods like one-dimensional histograms for automatic colour space selection [29] and two-dimensional histograms for improved segmentation [30] have been developed to optimise the process. Specific applications, such as YCbCr for face detection [31] and YIQ for satellite imagery [32], demonstrate tailored approaches. For monitoring the Auckland Harbour Bridge (AHB), the Y component (Equation (1)) from RGB to YCbCr transformation can enhance structural anomaly detection by focusing on luminance and minimising colour variations due to lighting or weather conditions.

$$Y = 0.299R + 0.587G + 0.114B \quad (1)$$

Techniques like histogram thresholding and SVM-based clustering have also been applied to enhance segmentation by structuring pixel data into coherent colour groups [33,34].

2.4. Advanced Techniques in Video and Image Analysis for ARDAD

In computer vision, object detection in videos involves recognising the movement of objects across multiple frames, utilising techniques like background subtraction, frame differencing, and optical flow. The Gaussian Mixture Model (GMM) is a prevalent method for background separation, enhancing foreground–background distinction by modelling pixel distributions with Gaussian mixtures [35–37]. Template matching, another practical approach, uses MATLAB 2022a functions like *normxcorr2* and *regionprops* to track objects in successive frames by identifying the peak of normalised cross-correlation [38]. Template matching, as represented by Equation (2), calculates the intensity-weighted centroid of an object using the following expression:

$$x_c = \frac{\sum_{i=1}^N x_i \cdot w_i}{\sum_{i=1}^N w_i} \quad (2)$$

where x_c is the centroid location, x_i is the pixel location, and w_i is the pixel intensity. Multiplying x_i by w_i in the numerator ensures that each pixel's location is weighted by its intensity, giving more importance to pixels with higher intensities in the centroid calculation.

Foreground–background separation techniques split a video into static backgrounds and dynamic foregrounds, using models that adapt to changes in the scene to detect motion and recognise objects [39,40]. Recent advancements in the GMM have introduced methods that handle complex scenes like traffic, improving the detection of slow-moving

or oversized vehicles [41]. Each pixel in the video sequence is modelled as a mixture of K Gaussian distributions. The probability of observing a pixel value X at time t is given by Equation (3).

$$P(X_t) = \sum_{k=1}^K \omega_{k,t} \cdot N(X_t | \mu_{k,t}, \Sigma_{k,t}) \quad (3)$$

Here, $\omega_{k,t}$ is the weight of the k^{th} Gaussian component at time t , N is the Gaussian distribution, $\mu_{k,t}$ is the mean of the k^{th} Gaussian, and $\Sigma_{k,t}$ is the covariance matrix of the k -th Gaussian. On the other hand, frame difference methods detect motion by comparing pixel differences from one frame to the next, a simple yet effective technique for identifying moving objects [42]. Optical flow techniques assess motion by analysing the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and the scene [43]. For example, Equation (4) synthesises motion detection using frame differences $|F_t - F_{t-1}|$ to identify changes and optical flow, represented by $\nabla I_t \cdot v_t$, to analyse motion patterns and speeds, with α adjusting their relative contributions.

$$\Delta M_t = |F_t - F_{t-1}| + \alpha \cdot \nabla I_t \cdot v_t \quad (4)$$

Object tracking in computer vision follows object detection, using colour, texture, shape, size, and orientation to track objects like vehicles or pedestrians across video frames. Robust tracking is essential across camera placements, lighting conditions, and cluttered scenes [44]. Techniques like the Kalman filter and particle filter are standard for tracking linear and non-linear motions [45]. Blob analysis tracks objects in binarised images based on features like area and bounding-box dimensions [46]. The Bayer filter pattern calculates the distance between object centroids, aiding in tracking and measuring vehicle velocity. The Kalman filter refines trajectory predictions by combining predicted and actual locations [47,48]. For non-linear scenarios, the Extended Kalman filter uses the Jacobian matrix for transition matrices (Equation (5)).

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - H_k \hat{x}_{k|k-1}) \quad (5)$$

In Equation (5), $\hat{x}_{k|k}$ is the updated state estimate at time k ; $\hat{x}_{k|k-1}$ is the predicted state estimate at time k , based on the estimate from time $k-1$; K_k is the Kalman gain, which dictates the blending of prediction with observation; z_k is the actual measurement at time k ; and H_k is the measurement matrix that maps the state space into the measurement space.

The particle filter enhances tracking in complex scenarios by sampling and resampling object features across frames, proving more effective in non-linear and cluttered environments than other methods [49,50]. It utilises a correlation particle filter to manage scale variations and feature interdependencies.

The mean shift algorithm, which uses a statistical colour model for tracking, iteratively converges to high-density areas in the colour space 8-point connectivity to locate objects [51]. The Adaptive Local Movement Model (ALMM) addresses movement by focusing on regional patches rather than the entire object, proving effective against occlusions and rapid movements [52].

Data collection and manual labelling in ARDAD systems can be resource-intensive and time-consuming. While deep learning approaches offer significant advantages in image and video processing due to their ability to outperform traditional methods, they require substantial datasets and computational power [53]. Alternatively, expert-driven feature extraction with traditional machine learning (ML) approaches can be effective with smaller datasets and less computational demand, making them suitable for initial target system designs [54]. Pretrained deep learning models facilitate automatic feature extraction and enable transfer learning, which adapts models to new functions without extensive computational resources [55] and catastrophic forgetting. The fusion of transfer learning with traditional machine learning enhances spatiotemporal classification by leveraging pre-trained models for efficient feature extraction and applying traditional models for sequence analysis, presenting an opportunity for improving classification accuracy (Figure 3).

Combining deep learning with traditional machine learning has been applied in various domains, such as traffic flow prediction and crop health monitoring [56,57].

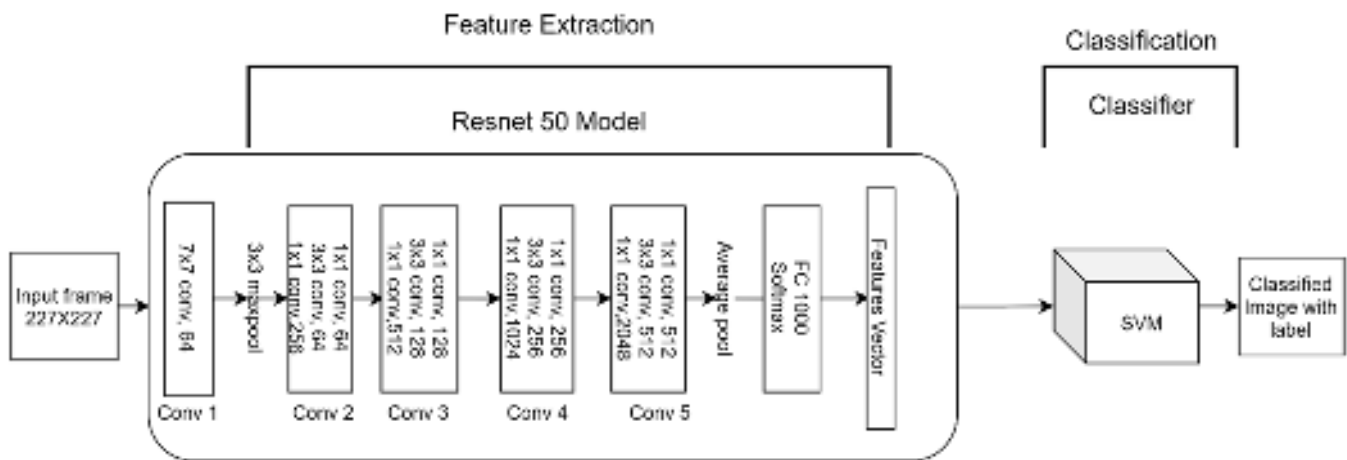


Figure 3. Integration of transfer learning with traditional ML: ResNet 50 extracts features from input images, which an SVM classifies. Integrating ResNet 50’s deep learning capabilities to generate feature space without relying on expert knowledge combined with a traditional classifier (e.g., SVM) presents the opportunity to enhance image classification performance.

The fusion of transfer learning with traditional machine learning (Figure 3) for spatiotemporal classification tasks is intended to replace the softmax() function (6), which is commonly used as the last layer of a neural network to convert high-dimensional feature vectors ($\in \mathbb{R}$) typically into a probability of possible outcomes, i.e., normalised for intended output class distribution.

$$\hat{y} = \text{softmax}(Wf_{\text{RNN}}(f_{\text{CNN}}(X)) + b) \quad (6)$$

where the loss function used in transfer learning is

$$L(\theta) = \frac{1}{N} \sum_{i=1}^N \ell(y_i, f(X_i; \theta)) \quad (7)$$

Here, $L(\theta)$ is the average loss over N training samples, X_i and y_i are the input and label for the i^{th} sample, f is the model with parameters θ initialised from a pretrained model, and ℓ is the loss function, such as cross-entropy loss for classification. In practical applications, such a hybrid approach enhances the efficiency and accuracy of spatiotemporal models in practical applications [56,57].

Key considerations from a project methodology standpoint include the availability and balance of datasets, the computational limitations of intended target platforms, and the integration of traditional ML techniques for enhanced data insights. The adopted methodologies encompass transfer learning for enhanced object detection and classification, leveraging deep learning for feature generation, and utilising traditional ML classifiers to extract further insights, ensuring a comprehensive approach to developing ARDAD solutions.

3. Materials and Methods

This paper documents a critical phase in the development process, transitioning from the initial Proof-of-Concept (PoC) [7] to a minimum viable product (MVP) for the Auckland Harbour Bridge’s movable concrete barrier (MCB), addressing additional research problems related to detecting anomalies in a noisy spatiotemporal scenario.

In addressing real-world challenges and producing technological solutions, the research methodology had to be significantly adapted to adhere to NZTA Waka Kotahi’s safety rules associated with the recording of videos showing pins deliberately set in tem-

porary unsafe positions (on the short barrier segments located on the safe working area before the Harbour Bridge). The data collection equipment is safely attached to a BTM using makeshift contraptions (Figure 4 and Table 1). However, leveraging multiple Barrier Transfer Machine (BTM) operations to collect sparse minority-class data was not viable, leading to the development of a background cloning approach to produce synthetic frames depicting unsafe pin positions (Table 2). Such a methodological approach is aligned with multidimensional problem solving for projects where ongoing adaptation to external factors is necessary [58,59].



Figure 4. Data collection setup: iPhone 13 Pro (for LiDAR), Samsung A7 Mobile, Apple iPad 6, External power bank, GoPro cameras and mounting equipment, and camera and iPhone mounting on a BTM.

Table 1. Data collection overview. Equipment, camera specifications, weather conditions, challenges encountered, and technological advancement progression across three data collection sessions.

Session No.	Equipment	Camera Specs	Weather Conditions	Challenges	Tech. Integration
1st	GoPro 8, GoPro 5, Samsung A7 Mobile, Apple iPad 6, duct tape, power bank, and mounting strips on the BTM. Barrier Transfer Machine (BTM) used for maintaining optimal camera angles during recording sessions	Resolution: 720p, Frame Rate: 240 fps, Field of View: Narrow, Audio: Wind Only, Protune: Enabled, White Balance: Auto, Colour: Flat, Shutter: Auto, ISO Limit: 6400, Sharpness: High, Audio Protune: Medium, Auto-rotation: Auto	Occasional rain and overcast	Vibrations, Camera Heating, Mounting Integrity Checks	Initial session: groundwork for incremental data collection
2nd	GoPro 8 and 5 on BTM's front and rear arms, other equipment same as first	Same as first session specs	Sunny	Camera Heating, Waterproofing, Battery Autonomy at High Frame Rates	The session focused on sunny and ideal lighting conditions

Table 1. Cont.

Session No.	Equipment	Camera Specs	Weather Conditions	Challenges	Tech. Integration
3rd	GoPro 9 and iPhone 13 Pro in hard case with 3D point cloud app on BTM's front arm, GPS enabled on both devices. iPhone 13 Pro utilized with a specialized application to create 3D point cloud data, enhancing the depth and quality of spatial analysis	GoPro 9: 1080p resolution, 240 fps, Upgraded specs for harsh conditions iPhone 13 Pro: Lidar-enabled camera for 3D point cloud data capture and GPS for location tracking	Early morning dark and rain	Ensuring Camera Waterproofing, Maintaining Battery Autonomy, Device Stability in Harsh Conditions	Integration of Lidar and GPS for advanced spatial data capture; utilised 5G for real-time data transmission. Use of 5G technology was considered to enhance real-time data capture and transmission, ensuring that large datasets could be managed effectively

3.1. Step 1: Data Collection and Synthetic Data Generation

Incremental data collection began with the first session under occasional rain and overcast conditions, setting a precedent for the resilience of the process. During the second session, conducted in sunny weather, two GoPro cameras were mounted on the arms of the Barrier Transfer Machine (BTM), with technical challenges such as vibrations, camera overheating, the need for waterproofing, and maintaining battery life at high frame rates becoming apparent. The third session was conducted in heavy rain, prompting further protocol and system design enhancements. Concurrently, integrating advanced technologies like Lidar, GPS, and 5G networks was considered to augment the robustness and capability of the data collection cycles (Table 1).

The pre-emptive manual inspections of moveable concrete barriers (MCBs) before any BTM operation for lane modification eliminated the possibility of naturally capturing the required unsafe pin positions (Figure 5). The precautionary measure is understandable given the substantial risk posed by potentially loose concrete blocks during the MCB transfer process, which could compromise the BTM's integrity and the safety of road users.



Figure 5. Pin manually pushed out of place by NZTA staff at the author's request. The process was challenging and labour-intensive, and the staff member needed several minutes per pin for adjustments. Note: Such an option was not viable as it does not represent the natural environment of the ROI where real-world problems need solving.

Table 2. Semi-automated process for generating a diverse dataset of synthetic pin positions (based on the manual cloning method illustrated in Figure 6).

Algorithm: Semi-Automated Synthetic Image Generation for Pin Position	
Input: Series of Pin_OK images, Pin_OUT template image	
Parameters:	
-Alignment normalisation method	
-Segmentation method	
-Displacement calculation method	
-Transformation method	
-Background reconstruction method	
-Edge refinement method	
Output: Synthetic dataset with varied and accurately positioned Pin_Out frames	
1: Load Images	
1.1: Load the series of Pin_OK images	
1.2: Load the Pin_OUT template image	
2: Normalise alignment	
2.1: Normalise the alignment of the pin in the series of Pin_OK images to a standard reference position	
3: Process each normalised Pin_OK image	
3.1: Segment the pin using a region-based segmentation method	
3.2: Calculate the displacement needed based on the Pin_OUT template	
3.3: Apply the calculated displacement to adjust the pin position and create a synthetic Pin_Out image	
3.4: Reconstruct the background where pin was initially placed	
3.5: Refine the edges of the moved pin to ensure seamless integration	
3.6: Visually validate the quality of the synthetic image	
3.7: If the quality is acceptable, add the synthetic image to the dataset	
3.8: If adjustments are needed, refine the parameters and repeat the process	
4: Repeat for all images	
4.1: Continue the process for all Pin_OK images in the series to create a comprehensive dataset with varied Pin_Out positions	

Faced with the sparsity of unsafe pin position, a novel synthetic data creation method involving background cloning from original video frames was introduced (Figure 6). Cloning allows for generating varied representations of the minority class, thus addressing the skewed data distribution. Initial attempts yielded imperfect frames with jagged edges, but the process was honed through iterative refinement, resulting in high-fidelity synthetic frames that significantly bolstered the dataset. Adding the synthetic frames facilitated the fine-tuning of model precision and recall, which is crucial for the efficacy of automated inspection systems and ensures higher sensitivity towards false negatives—a priority for traffic safety systems. The cloning approach proved pivotal in circumventing the limitations imposed by manual data collection methods, paving the way for a safer, more efficient means of training robust detection models.

As shown in Figure 6, creating synthetic frames using an image editor takes at least 20 min per frame, highlighting the need for a more efficient and automated solution (Table 2). The process begins by loading a series of images where the pin is in a safe position (Pin_OK) and a template image where the pin is in an unsafe position (Pin_OUT). The method involves several key steps: normalising the alignment of the pins in each Pin_OK image to a standard reference position, segmenting the pins using a region-based segmentation method, and calculating the displacement needed to replicate the Pin_OUT template.

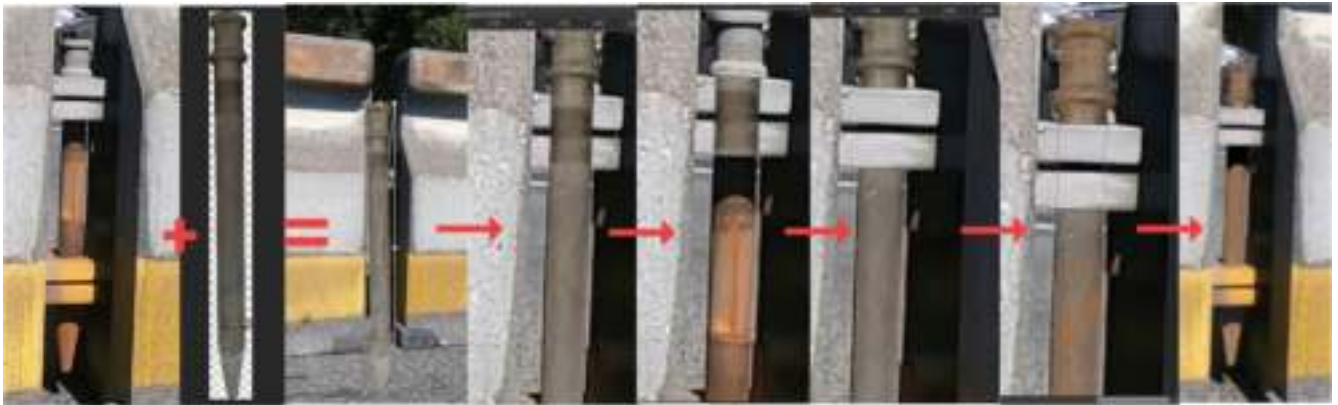


Figure 6. Synthetic data generation: The image illustrates the various stages of manipulating a video frame to create synthetic images depicting a metal pin in unsafe positions. Starting with a standalone image of the metal pin, the process involves adjusting its orientation, position, and environmental context to generate realistic, unsafe scenarios.

3.2. Step 2: Data Augmentation

Data augmentation was necessary to work with sparse and unbalanced datasets. We also considered commonly used approaches to enhance the training robustness of neural networks by generating additional training data from existing datasets and helping to prevent overfitting [60]. The augmentation techniques applied included geometric and affine transformations where rotation, resizing, reflection, translation, and shearing were utilised to modify the image structure without altering its content [37] (Figure 7). The transformations were implemented using MATLAB's Image Processing Toolbox, which facilitated the efficient application of such techniques. Additionally, this research incorporated methods to introduce realistic variations into the dataset by adding noise and blur effects, specifically using Gaussian and Salt and Pepper noise patterns. Such effects are applied using MATLAB's `imnoise` function for noise addition and `imgaussfilt` for Gaussian blur. Such modifications simulated potential real-world imperfections in data, aiding the network in learning to handle such irregularities effectively.

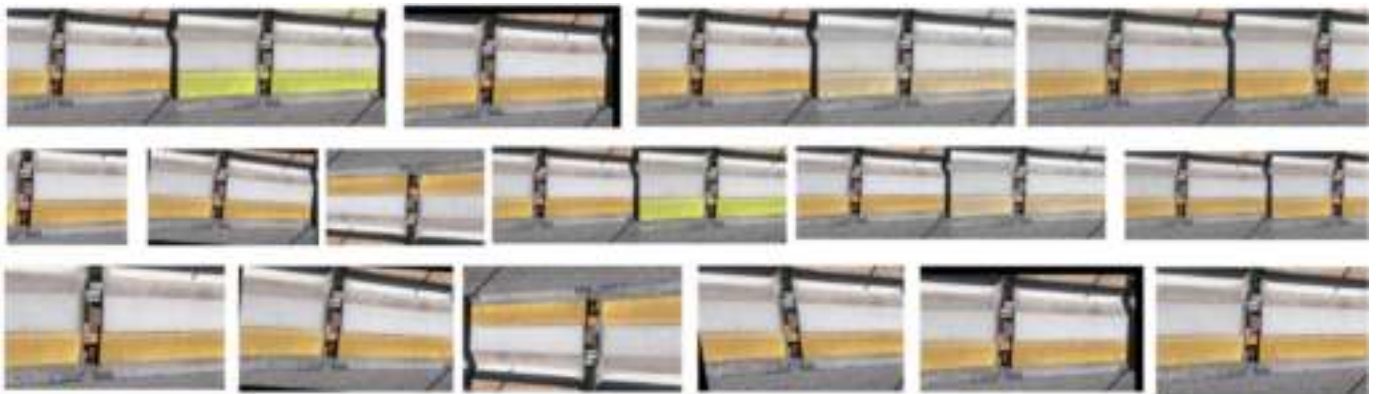


Figure 7. Illustrative outputs from data augmentation techniques showcase the range of image transformations applied to enhance model training. It includes geometric transformations such as rotation and scaling, affine transformations with various translations and shearing, and visual effects such as Gaussian blur, noise addition, and sophisticated colour adjustments. The modifications are instrumental in preparing the neural network to handle diverse and realistic scenarios encountered in practical applications, Xu, Yoon [61].

Further innovation includes the adoption of advanced colour transformation techniques. Using MATLAB's `jitterColorHSV` function, the images underwent random adjust-

ments in brightness, contrast, hue, and saturation, broadening the range of visual data the model was exposed to during training. Additionally, a novel colour transformation approach that combines table lookup methods and 3D colour space interpolation allowed for high-quality, real-time colour processing.

The comprehensive approach to data augmentation diversifies the training set, enhancing the model's generalisation capabilities, which are important for similar applications, including smart city contexts and the application of technology to improve the usability of spaces where human activity may occur.

3.3. Step 3: Data Distribution Analysis Post Minority Boosting

Synthetic frames balanced the dataset, enhancing classification accuracy on a relatively small, manually labelled dataset (Figure 8). This approach was advantageous under restrictive conditions restricting minority class data collection. Additionally, synthetic data facilitated fine-tuning model performance metrics such as Precision and Recall, focusing on minimising false negatives due to their criticality for safety.

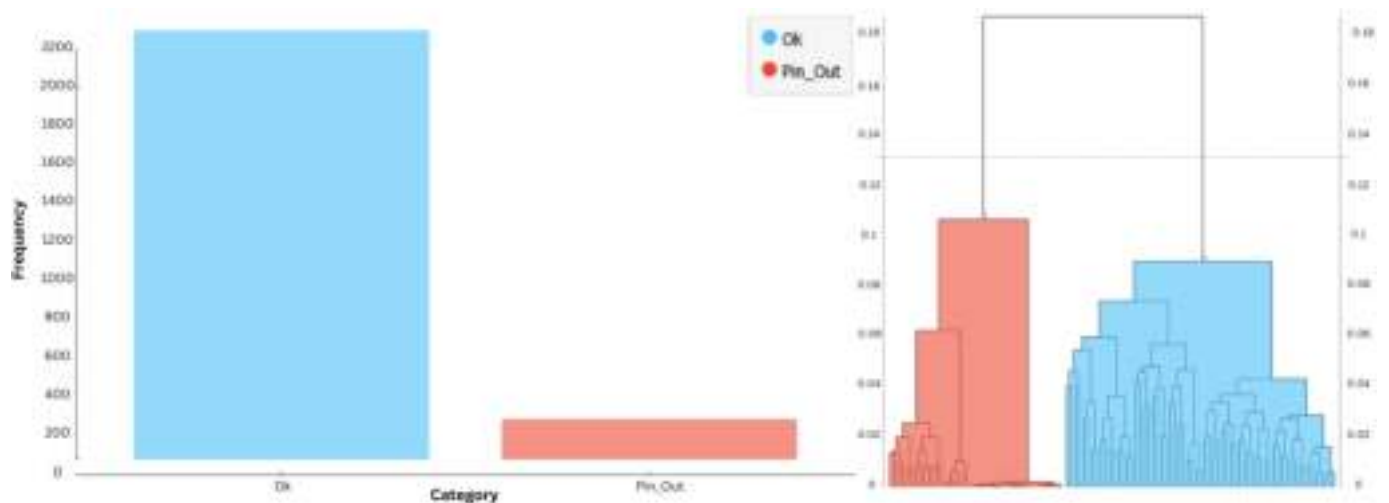


Figure 8. The dendrogram graph, derived from the data shown by the bar graph, shows two clusters (Pin_OK (red) and Pin_Out (blue)) and a visual separation of the generated multidimensional feature space.

From approximately 30,000 frames, 2300 images showing safe pin positions were curated. To address the challenge of limited occurrences of unsafe pin conditions—a common issue in training robust detection systems—210 synthetic images simulating unsafe pin positions were generated and added to the dataset (Table 3).

Table 3. Data distribution: overview of the number of Pin_OK and Pin_Out images used for training and validation across different video recording sessions.

Training Video Session No. (Table 1)	Training Data		Validation Data	
	Pin_OK Images	Pin_Out Images	Pin_OK Images	Pin_Out Images
1	155	32	40	5
2	725	80	180	20
2	800	40	200	10
3	400	16	100	5

Creating varying degrees of ‘Pin_Out’ positions helped optimise the balance between false positives and negatives, which is crucial for refining the model’s accuracy and robustness (Table 4).

Table 4. Initial classification results achieved from data clusters from Figure 8. Adopted from Bačić, Rathee [7].

Confusion Table				
Model			Actual	
			PIN_OK	PIN_OUT
Logistic regression	Predicted	PIN_OK	[98.5%	1.5%
		PIN_OUT	0	100%]
PIN_OK		[98.9%	1.1%	
PIN_OUT		0	100%]	
PIN_OK		[98.1%	1.9%	
PIN_OUT		0	100%]	
SVM				

Synthetic frames varying in Pin_Out positions improved the tuning of false positive (FP) and false negative (FN) ratios. The aim is to achieve $FP > FN$ using a dataset with borderline unsafe pin positions, enhancing classification accuracy. Accuracy was measured using the following formula:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (8)$$

TP represents the true positives or correctly identified Pin_Out frames, and TN represents the true negatives or correctly identified Pin_OK frames. Precision, the ratio of correctly predicted positive observations to the total predicted positives, is defined as follows:

$$\text{Accuracy} = \frac{TP}{TP + FP} \quad (9)$$

The comprehensive approach involved a meticulous three-fold stratified cross-validation process, as detailed in Table 5.

Table 5. The cross-validation test and score results updated from initial research.

3-Fold Stratified Cross-Validation		
Model	Precision	Recall
Logistic regression parameters: Regularisation: Ridge (L2), C = 1	0.995	0.995
Multilayer Perceptron (MLP)- Parameters: Hidden layers: 2 Neurons: Activation function: ReLu Solver: Adam Alpha: 0.02 Max. iterations: 200 Backpropagation algorithm	0.995	0.995
Support Vector Machine (SVM) parameters: C = 1.0, ϵ = 0.1 Kernel: Linear Numerical tolerance: 0.001 Max. iteration: 100	0.985	0.984

3.4. Step 4: Detecting Region of Interest (ROI)

Our research team comprehensively investigated various Region of Interest (ROI) detection methodologies in developing an automated pin detection system for traffic management applications. Each method was scrutinised for its ability to accurately identify pin locations within video frames under various environmental conditions. Shuffling through different methodologies illuminated each approach's challenges and potential, culminating in us finding a particularly effective solution. Initially, our exploration utilised region proposal methods, which calculate bounding boxes around potential ROIs (Figure 9C). While initially promising, the approach necessitated frequent manual adjustments to ensure precision in detection and counting, rendering it less feasible for dynamic real-world applications where automation is paramount. A Gaussian Mixture Model (GMM) was employed for adaptive background modelling (Figure 9B). The GMM was adept at segmenting moving pins from static backgrounds by modelling the probability density distribution of pixel intensities. Despite its effectiveness in foreground differentiation, the GMM required extensive post-processing to localise each pin accurately, limiting its standalone utility in precise ROI detection. We also explored colour-based segmentation using K-Means clustering within RGB, LAB, and HSV colour spaces—the latter of which is particularly favoured due to its resilience against variations in lighting (Figure 9A). The method exploited the unique colour signatures of the pins but faced limitations in specificity, often capturing unrelated objects sharing similar colour profiles.

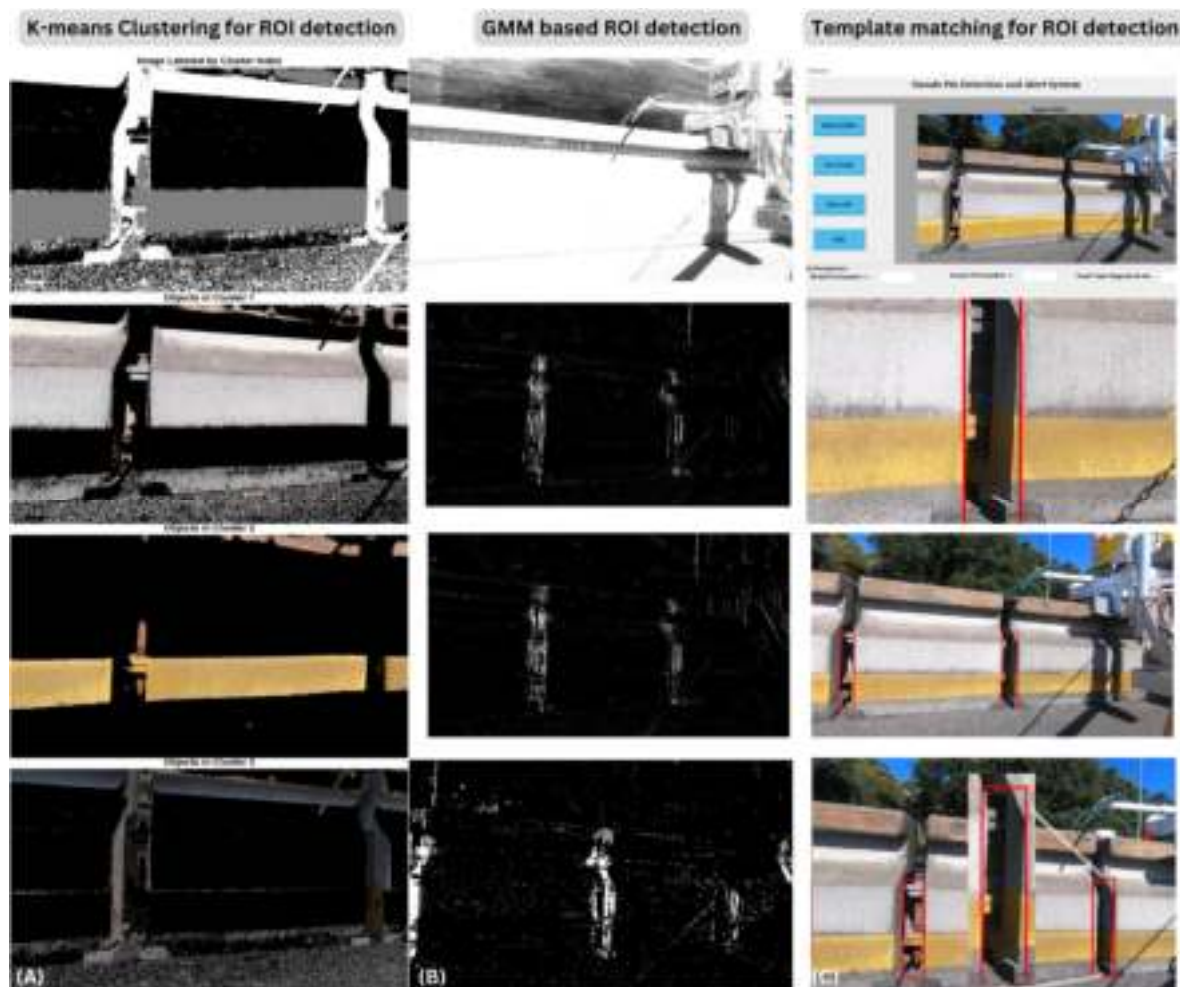


Figure 9. Comparative analysis of promising ROI detection techniques in automated pin detection: (1) colour-based segmentation using K-Means clustering in HSV colour space, highlighting the unique

colour signatures of pins while addressing challenges in specificity. (2) Gaussian Mixture Model (GMM)-based detection illustrates foreground–background segmentation and pin movement tracking between concrete blocks. (3) Regionprops application, showcasing template matching and automated bounding box detection for precise localisation and reduced manual intervention in pin ROI detection and labelling.

The methods tried in Figure 9 led to us adopting MATLAB's *regionprops* function, which proved superior in addressing previous methods' precision and automation challenges. The *regionprops* analyses each connected a component in binary video frames, effectively quantifying the area and bounding box coordinates for each detected ROI. The function enhanced the accuracy of pin detection and facilitated a significant reduction in manual intervention by automating the ROI detection process.

The underlying mathematical concepts of the *regionprops* function are integral to its performance (Equations (8)–(13)).

Area: The area of a region is computed as the number of pixels within it.

$$\text{Area} = \sum_{(i,j) \in R} 1 \quad (10)$$

In this equation, the area represents the total number of pixels in the region R. Each pixel within the region contributes a value of 1 to the total area, effectively counting the pixels.

Centroid: The centroid is determined by averaging the positions of all the pixels in the region, as follows:

$$\text{Centroid}_x = \frac{1}{N} \sum_{(i,j) \in R} i, \text{Centroid}_y = \frac{1}{N} \sum_{(i,j) \in R} j \quad (11)$$

Bounding box: The bounding box, which encloses the region, is defined by the minimum and maximum x and y coordinates of the pixels.

$$\text{BoundingBox} = [x_{\min}, y_{\min}, x_{\max} - x_{\min} + 1, y_{\max} - y_{\min} + 1] \quad (12)$$

Major and minor axis length: For regions approximated by ellipses, the major and minor axis lengths are derived from the eigenvalues of the covariance matrix of the pixel coordinates.

$$\text{MajorAxisLength} = 2\sqrt{\frac{\lambda_1}{N}}, \text{MinorAxisLength} = 2\sqrt{\frac{\lambda_2}{N}} \quad (13)$$

Orientation: The orientation of such an ellipse, indicating the angle between the x-axis and the major axis, is calculated using the second moments of the region.

$$\text{Orientation} = \frac{1}{2} \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \quad (14)$$

Perimeter: The perimeter is measured by summing the distances between consecutive boundary pixels.

$$\text{Perimeter} = \sum_{\text{boundarypixels}} \text{distance between consecutive boundary pixels} \quad (15)$$

The integration of *regionprops* into our method significantly advances the automation of pin detection. The enhancement offers high precision and efficiency, which are critical for traffic management systems requiring reliability and rapid processing.

Our successful implementation emphasises the importance of a methodical approach in engineering research, demonstrating how iterative testing and the integration of various techniques can tackle complex real-world problems. This study highlights the effectiveness of combining traditional image processing methods with advanced analytical tools and sets a new standard for future research in automated traffic safety and management systems.

The findings provide a robust framework for researchers and practitioners to enhance automated detection systems across engineering domains.

3.5. Step 5: Model Training and Validation

The computing system (Table 6) provided sufficient hardware capabilities to train the model at the core of the developed ‘Unsafe Pin Detection and Alert System’ system.

Table 6. A development system using NVIDIA GPU parallel processing architecture.

System Configuration	
Processor	Intel Core i7 Processor
Memory	32 GB RAM
Hard Drive	512 GB Solid State Drive
Graphics	NVIDIA GeForce RTX2070 Super 8 GB GFX
Operating System	Windows 10

ResNet-50, enhanced with 3D convolutions within the STENet framework, was selected after an extensive comparison with other models like Squeezenet, GoogLeNet, InceptionV3, and Mobilenetv2. ResNet-50 is favoured due to its optimal balance between computational efficiency and performance, confirmed by satisfactory GPU RAM evaluations during initial experiments (Table 7).

Table 7. Comparison of pretrained deep neural networks based on input image resolution, number of parameters, depth, and model size.

Pretrained Deep Neural Networks				
Model	Input Image Resolution	Parameters (1,000,000)	Depth	Size
AlexNet	227 × 227	61	8	227
SqueezeNet	227 × 227	1.24	18	4.6
GoogLeNet	224 × 224	7	22	27
Inception v3	299 × 299	23.9	23.9	48
MobileNet v2	224 × 224	3.5	3.6	53
Resnet 50	224 × 224	25.6	50	96

Various environmental elements, such as rust-coloured foliage, closely matched the appearance of rusted metal pins, making visual differentiation challenging. Similarly, unrusted metal pins shared the same colour as the road surface and the tyres of passing vehicles, further complicating the identification process. The metal barriers in the background often had shapes resembling the horizontal profile of the metal pins, while the dynamic colours of passing vehicles introduced intense background variations. In addition to labelling the pin statuses, regions of interest (ROIs) were defined on each image using the *regionprops* method to distinguish between relevant features and potential background noise. This method, which analyses each connected component in binary video frames to quantify the area and bounding box coordinates for each detected ROI, significantly enhanced the accuracy of pin detection and reduced manual intervention. The combined consideration of shape and colour homogeneity was a significant factor in training the AHB pin detection model (Figure 10).

Additionally, certain parts of the movable concrete barrier (MCB) system closely resembled the shape of the metal pins, creating further confusion (Figure 11(3)). These factors made the detection task challenging, as the system had to differentiate between multiple visually similar elements under varying conditions. The dataset then underwent a series of image augmentation processes to simulate various operational conditions not covered by the initial video capture. The processes included geometric transformations

such as rotations, scaling, translations, and adding image perturbations like Gaussian blur and noise, which are crucial for training models to perform well under practical deployment conditions.

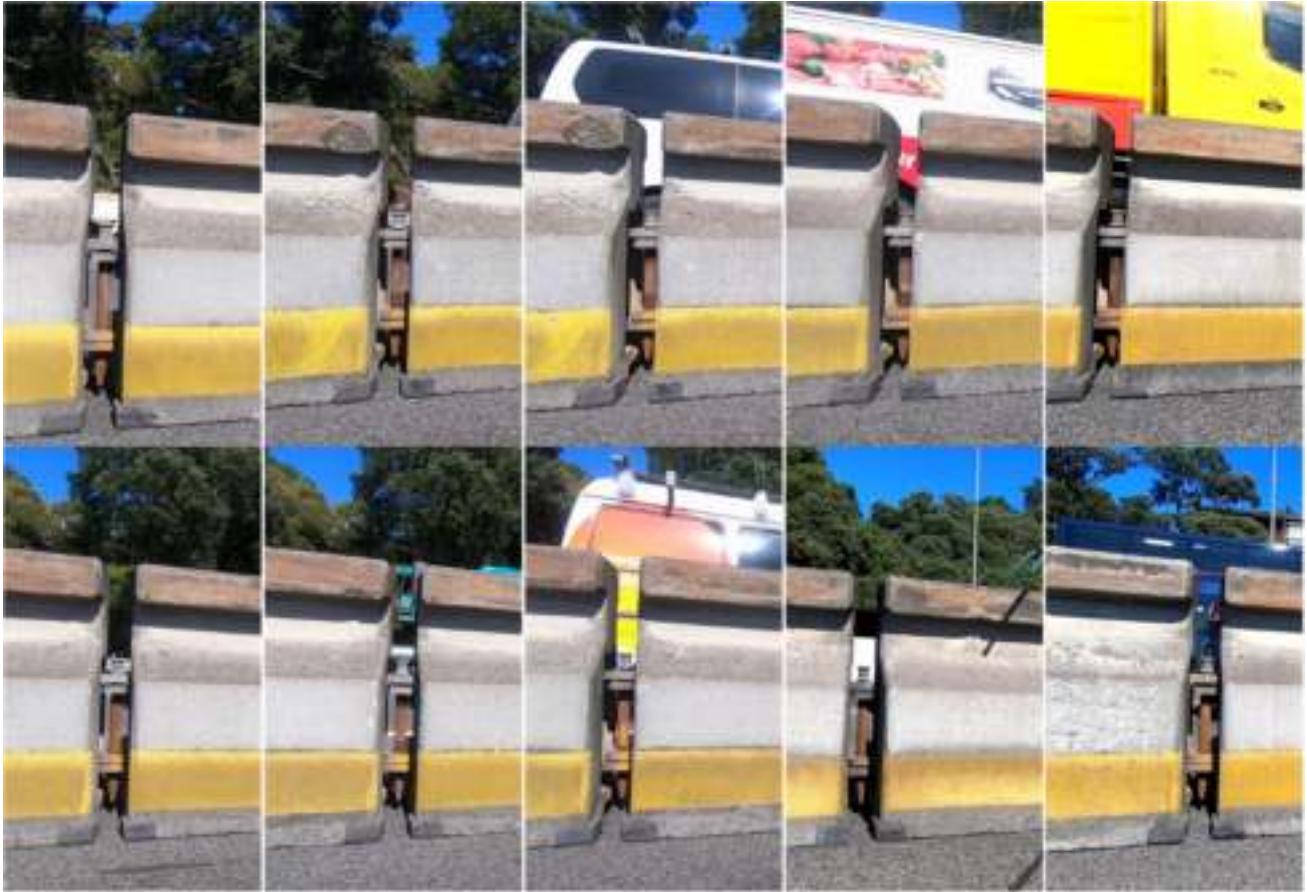


Figure 10. Illustration of the background challenges in ROI detection.

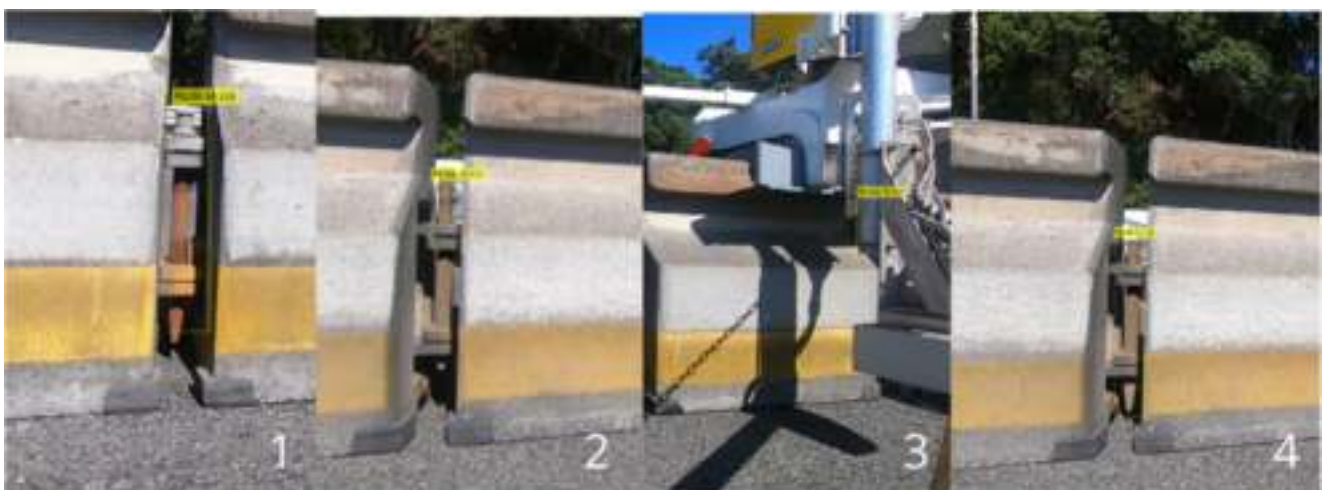


Figure 11. Pin detection examples mixed with false-positive detection (3) illustrate a high level of background noise that looks similar to features of the ROI.

A critical component of the training process was the adjustment of the learning rate, a parameter that significantly impacts the convergence and final performance of the model.

Learning rates ranging from 0.01 to 0.00001 using ResNet-50 were tested (Figure 12). The optimal rate was determined through a series of trials evaluating model accuracy and loss metrics. The experiments utilised ROC curves to visually represent the trade-offs between true-positive and false-positive rates at various threshold settings, enabling an informed selection of the best-performing model under the given training conditions. The results consistently showed that a learning rate of 0.0001 provided the best balance between training speed and model accuracy.

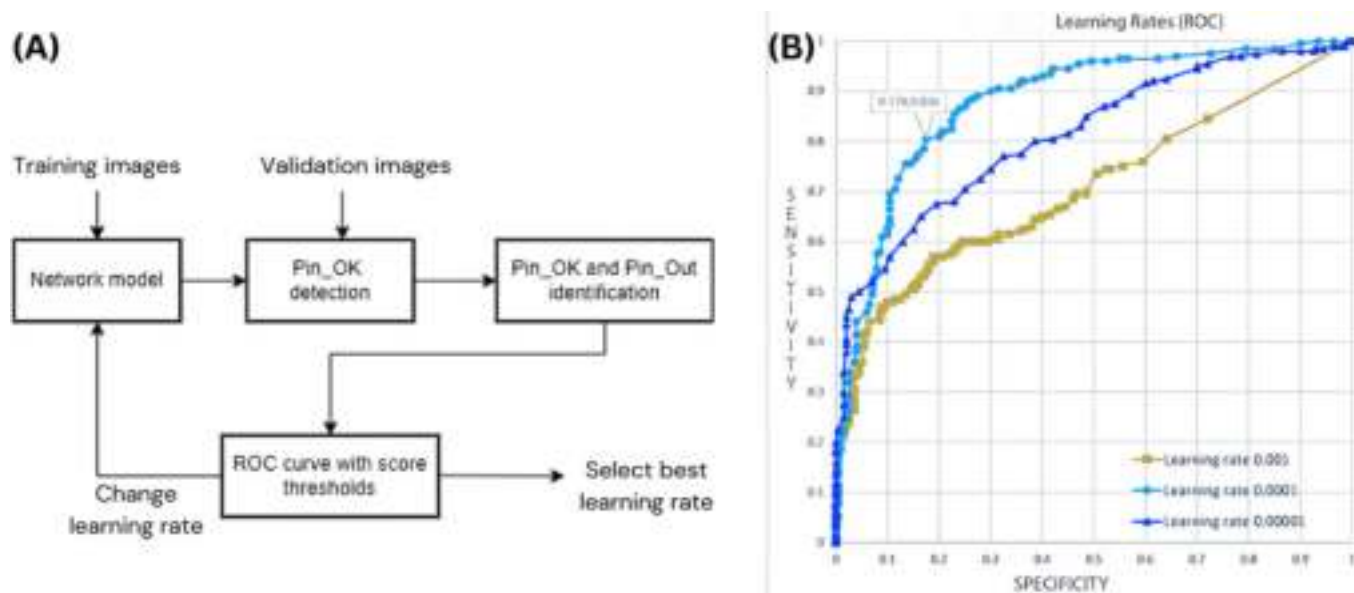


Figure 12. The training process to find the best network model based on learning rate: (A) flow of the training process and (B) ROC curves for different learning rates showing the performance of learning rates. The learning rate of 0.0001 was chosen as it balances training speed and accuracy, providing better performance while avoiding the high computational cost and time.

In conclusion, this research detailed the selection, preparation, and training of a deep learning model capable of detecting and classifying metal pin statuses in movable concrete barriers. The methodological approach, emphasising the creation of a comprehensive and diverse training dataset through real and synthetic images and rigorous model training protocols, demonstrates a scalable and efficient solution to enhance public infrastructure safety through advanced AI techniques.

3.6. Step 6: Model Analysis and Testing

The STENet architecture, inspired by and building upon the robust framework of ResNet-50, exhibits considerable potential for spatiotemporal anomaly detection tasks, such as those needed for monitoring the Auckland Harbour Bridge (AHB). While ResNet-50 was initially developed for image classification, its strong spatial feature extraction capabilities make it a solid foundation for further enhancements to analyse video and live feed data. STENet integrates components from ResNet-50 with 3D convolutions to capture spatial and temporal information, facilitating the direct analysis of motion and dynamic changes within video sequences. Further enhancements involve leveraging ResNet-50's spatial feature extraction capabilities and Recurrent Neural Networks (RNNs) to track temporal sequences or employing temporal pooling methods to summarise video segments efficiently.

By adopting strategies for model simplification, edge computing, and incremental learning, STENet enhances system efficiency and responsiveness to new anomalies. This makes STENet particularly suited for continuous surveillance and traffic monitoring applications. Table 8 highlights the performance metrics of STENet compared to other models. By incorporating adaptations of ResNet-50's core features, STENet effectively recognises

appearance and behavioural pattern deviations over time, providing a robust framework for real-time anomaly detection across diverse operational environments (Figure 13).

Table 8. The Spatio-Temporal Enhanced Network (STENet) achieves an outstanding 95.2% accuracy score and an F1-Score of 94.8%, showcasing its exceptional capability to navigate through the challenges of background noise and small ROIs, with a remarkable ROC-AUC of 98.5%, solidifying its robustness in class differentiation in spatio-temporal tasks.

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Training Time	Inference Time	Number of Parameters
STENet	95.2%	94.5%	95.2%	94.8%	98.5%	4 h	80 ms	25 M
VGG-19	88.0%	88.5%	89.0%	88.7%	94.0%	6.5 h	90 ms	142 M
ResNet-50	90.0%	90.3%	91.0%	90.6%	95.8%	4 h	70 ms	25.6 M
InceptionV3	91.0%	91.2%	91.5%	91.3%	96.0%	3.5 h	65 ms	23.8 M
SqueezeNet	85.0%	85.5%	86.0%	85.7%	92.0%	2.5 h	35 ms	1.25 M

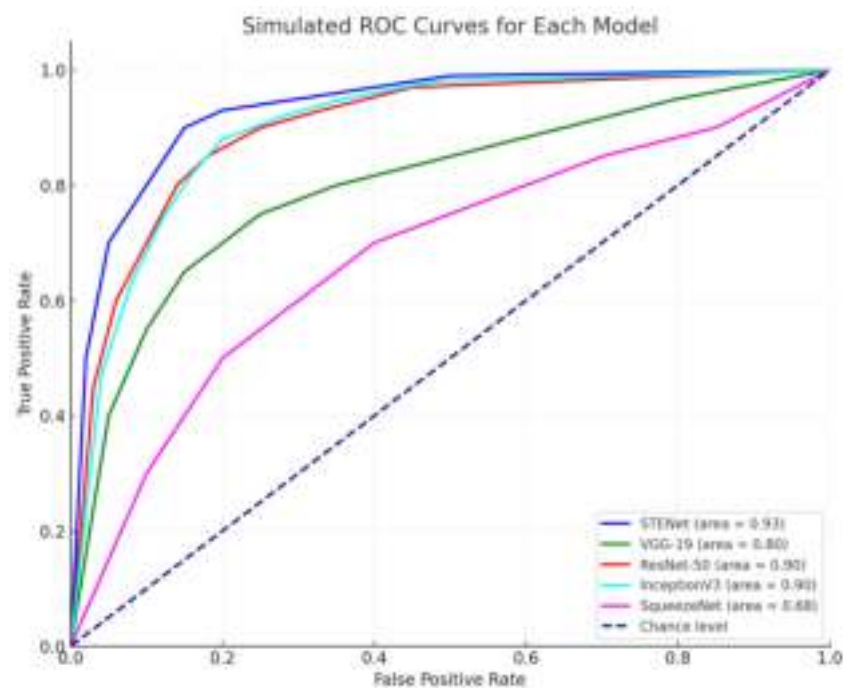


Figure 13. ROC curve comparison of Table 8.

4. Results and Discussion

The efficacy of the pin detection and alert system, which encompasses both pin Region of Interest (ROI) detection and tracking alongside pin status detection and alert functionalities, was critically evaluated. The assessments drew on methods previously detailed, with particular attention to the operational conditions influencing system performance.

4.1. Preparations, Data Recording Protocol, Findings, and Insights

The system's performance was assessed by examining recorded videos frame by frame, focusing on pin ROI tracking. Several environmental and operational factors were identified that could adversely affect the results.

- **Lighting Conditions:** Overcast conditions significantly hampered visibility, challenging the detection of pins between concrete blocks and rendering the detection and counting processes unreliable.

- **Background Movements:** Moving vehicles introduced significant noise into the background, severely hindering detection accuracy. While the greyish hue of the road and concrete blocks helped simplify the background, the large vehicles passing by disrupted visual clarity. Additionally, the rust colours and yellow of the movable concrete barriers (MCBs) and fragments of broken concrete barriers and metal barriers often blended into the pin regions of interest (ROIs), further decreasing tracking accuracy.
- **Shadows and Lighting:** Shadows cast on pin ROIs were sometimes misinterpreted as moving objects. Darker shadows could compromise the detection and tracking systems, particularly under bright sunlight.
- **Vibrational Distortions:** Operational speeds exceeding 6 km/h induced excessive vibration, blurring the images and introducing background noise, further complicating the detection processes.

To mitigate weather-related variabilities, multiple videos recorded in sunny conditions were analysed. The first video, capturing a broader field from the front arm of the Barrier Transfer Machine (BTM), displayed approximately two ROIs per frame, shot during a clear afternoon. The second video, taken from the BTM's rear arm, showed a narrower field of view with sometimes only partial visibility of a single-pin ROI.

4.2. Creating Synthetic Frames

The concept of creating synthetic frames encountered significant challenges. The initial cloning techniques were insufficient, yielding a non-viable minority dataset, which necessitated a shift from classification techniques to traditional mathematical approaches. Subsequent consultations with graphic experts led to the adoption of advanced methods using Photoshop and Gimp for generating synthetic frames (Figure 14).

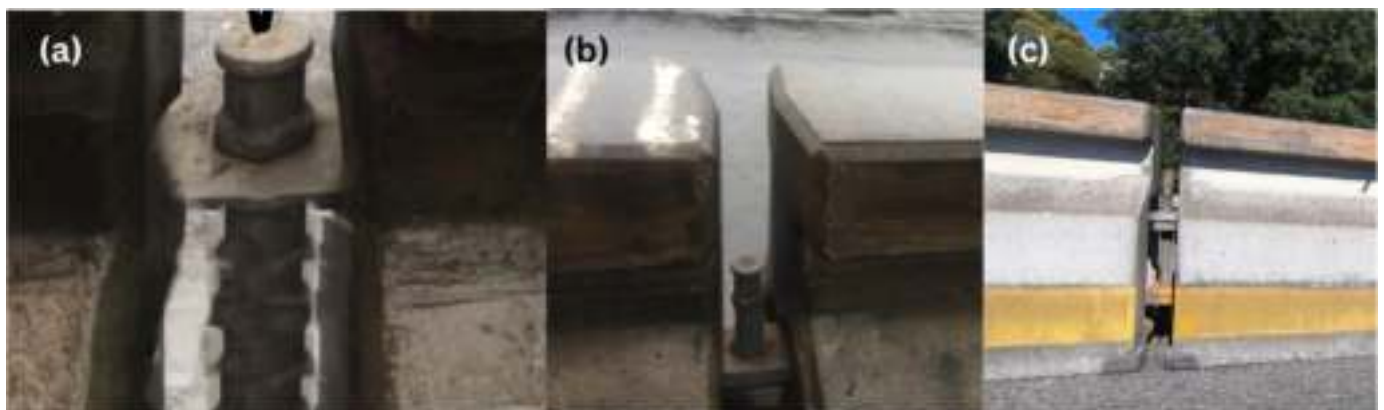


Figure 14. Cloning process to create synthetic frames depicting the 'PIN OUT' position. (a) Failed attempts until success (b,c); two distinct angles: (b) camera mounted at the back arm of the Barrier Transfer Machine (BTM) and (c) camera mounted at the front arm of the BTM for enhanced modelling accuracy.

Approximately 40 synthetic frames were crafted from video footage captured on a mobile phone during the Proof-Of-Concept phase. A second batch of 200+ synthetic frames was later produced from higher-quality video recordings, demonstrating a noticeable improvement in frame clarity between the initial and later productions. Given the labour-intensive nature of manual frame creation, efforts were made to automate the process using Photoshop's action panel. However, the complexity of the background elements hindered full automation. While MATLAB offers pixel cloning techniques, further research is required to develop a fully automated and robust method for synthetic frame generation. The analysis underscores the critical dependencies of environmental conditions on the performance of pin detection systems and highlights the ongoing need for

technical enhancements in synthetic data generation to support robust system training and validation.

4.3. System Prototype: Pin Tracking, Counting, and Alerting Functionality

The development of the metal pin detection and alert system was guided by the requirement that the end-user, presumed non-technical, has a simplified interface for efficient system operation. The system prototype, named ‘Unsafe Pin Detection and Alert System’ (Figure 15), was designed and implemented using MATLAB App Designer [62], integrating back-end operations with a graphical user interface (GUI).



Figure 15. Graphical user interface for ‘Unsafe Pin Detection and Alert System’.

System Functionalities

The App allows the end-user to upload a video or frame sequence for analysis. Upon upload, the App leverages a deep learning-based pin status detection network to analyse the video feed. The system maintains a count of metal pins, and if it detects a pin in an unsafe position, it triggers an alert and displays the specific metal pin ROI on the screen. The second functionality supports live feed analysis, where the user can connect the App to a live camera feed. The pin detection network activates to analyse incoming frames in real-time as the live feed is transmitted to the Unsafe Pin Detection and Alert System monitoring App (Figure 15). Like the pre-recorded analysis, the system tracks and counts the metal pin ROIs, alerting the user and displaying the ROI number when an unsafe pin is detected. The metal pin detection and alert system interface, as shown in Figure 15, is designed to be intuitive and easily navigable, ensuring that users can operate the system without prior technical knowledge. The App’s design and operational logic were specifically tailored to meet the needs of NZTA, allowing for custom modifications to accommodate specific operational requirements or updates.

In conclusion, in theory, the alert system App is ready to be deployed. There is potential for further development into a complete product deployment, pending interest and additional support from the NZTA. Future work will focus on refining the App’s functionality, enhancing its performance, and ensuring its onsite installation and integration into existing infrastructure systems. Such enhancements, however, depend on the availability of further resources and continued interest from stakeholders.

4.4. Results

The evaluation of the performance of the STENet model was aimed at effectively finding and localising the Pin_Out status using the workstation specifications outlined previously in Table 6. We applied the models to the pin's region of interest (ROI) frame-by-frame tracking and real-time counting, assigning each pin ROI an index to maintain continuity across frames. The detector model configurations are summarised in Table 9, which details parameters such as batch size, number of epochs, learning rate, and optimiser choices. Two optimisation techniques were utilised, Stochastic Gradient Descent with Momentum (SGDM) and Adam, both chosen for their ability to enhance model performance through efficient direction finding in the optimisation landscape [63,64].

Table 9. Configuration of the detector model. The table lists the batch size, number of epochs, learning rate, and optimiser used for training the detector model. Two different optimisation techniques, Stochastic Gradient Descent with Momentum (SGDM) and Adaptive Moment Estimation (Adam), were employed to enhance model performance.

Model	Batch Size	Epoch	Learning Rate	Optimizer
Detector	64	100	0.0001	Sgdm
	64	100	0.0001	Adam

Pin ROI tracking was evaluated for accuracy by comparing the consistency of index assignment from frame to frame. Discrepancies in index continuity were noted as errors in tracking. The dataset was divided into 70% training frames, 10% validation, and 20% testing frames to assess the model robustly across varied conditions. The training process using SGDM achieved quicker training times than Adam, as shown in Table 10. The validation of the models demonstrated lower Root Mean Square Error (RMSE) and validation losses with SGDM, indicating a more efficient optimisation path.

Table 10. SGDM training and validation process.

Epoch	Iteration	Time Elapsed	Mini-Batch (RMSE)	Validation (RMSE)	Minibatch Loss	Validation
10	150	00.15.05	0.91	0.87	0.8285	0.7910
20	300	00.30.20	0.74	0.83	0.5384	0.6761
30	450	00.45.12	0.66	0.78	0.4141	0.6329
40	600	01.00.15	0.62	0.76	0.3968	0.5809
50	750	01.14.50	0.56	0.77	0.3182	0.4150
60	900	01.28.58	0.53	0.76	0.2708	0.4060
70	1050	01.44.55	0.52	0.74	0.2408	0.3716
80	1200	01.58.21	0.50	0.76	0.2018	0.3208
90	1350	02.13.43	0.48	0.75	0.1808	0.2710
100	1500	02.27.23	0.47	0.74	0.1280	0.2219

The results of the pin ROI classification and the accuracy of bounding box detection by the trained STENet are depicted in Figure 16. The system was highly effective in recognising both Pin_OK and Pin_Out statuses, even when pin ROIs were partially obscured or closely positioned, which traditionally challenges detection accuracy. However, the accuracy diminished in frames where pins were too closely spaced or partially out of the field of view.



Figure 16. The selected pin ROI video frames showing generated overlays with bounding boxes.

While the system demonstrated robust performance in ideal viewing conditions, the detection accuracy varied under different fields of view and lighting conditions, emphasising the need for a more diverse training dataset. The creation of synthetic frames to augment the dataset was explored, but automation of the process remains a challenge for future work, as manual frame creation proved time-intensive. Table 11 showcases the classification metrics—accuracy, precision, and recall—after training, highlighting the model’s strong performance overall.

Table 11. Performance of the STENet where ResNet 50 is used as a classifier.

Classes	Accuracy		Precision		Recall	
	Training	Validation	Training	Validation	Train	Validation
Pin_Ok	0.952	0.945	0.942	0.940	0.940	0.945
Pin_Out	0.925	0.922	0.910	0.900	0.900	0.890

Despite variations due to the limited diversity in the minority class data, the classifier maintains high accuracy and precision across both classes. The consistent performance metrics for the ‘Pin_Ok’ and ‘Pin_Out’ classes demonstrate the model’s robustness and reliability in identifying and classifying majority and minority class instances.

The results underline the practical application of deep learning models to metal pin detection tasks, highlighting the necessity for further improvements in model training and

synthetic data generation to handle diverse operational scenarios effectively. The research's main contributions to ARDAD are listed as follows:

- Spatiotemporal analysis for automated monitoring of traffic barriers on the Auckland Harbour Bridge and other traffic locations using the same barrier to control traffic flow;
- Transforming a PoC [7] into an MVP with deployable AI algorithms for real-time ARDAD, exemplifying the translation of research into practice;
- Semi-automated synthetic data generation methods to enhance machine learning models for complex ARDAD tasks, addressing critical traffic events' data sparsity and rarity;
- Integrating machine learning with kernel manipulation for dynamic anomaly detection to improve the precision of current ARDAD systems, increasing the average detection accuracy from 0.826 to 0.939;
- Engaging in interdisciplinary collaboration to align ARDAD advancements with stakeholder requirements, merging computational research with traffic management solutions.

4.5. Discussion

The initial attempts to record high-speed videos from public transportation and personal cars presented significant challenges. Traffic flow predictions indicated that speeds could occasionally drop below 20 Km/hr on the Auckland Harbour Bridge, ideally allowing for capturing frames at 240 fps showing perfect pin alignments. However, these conditions were rarely met, and reliance on traffic jams during rush hours did not yield the desired outcomes due to erratic stoppage times and limitations of the recording equipment. Given the critical safety requirements on the Auckland Harbour Bridge, all data collection efforts were supervised by NZTA experts, who also provided access to the Barrier Transfer Machine (BTM) and the operation site, along with necessary safety briefings. To see the narrow gaps between the movable concrete segments, high-frame-rate cameras (GoPro 5, GoPro 8, and GoPro 9) were mounted on the BTM, which moved between 6 and 9 Km/hr. Recordings were made under various weather conditions and times of the day to capture diverse operational scenarios. The difficulty in finding and recording pins that were out of position significantly hindered the research process. After numerous unsuccessful attempts, synthetic frames were adopted as a viable solution. An interim report was provided to the NZTA, showcasing hierarchical clustering and a visual separation of feature vectors related to the minority output class using Pearson and Cosine correlation-based distance measures. Such computationally more demanding measures were selected over simpler ones like Euclidean due to the high dimensionality of features extracted from CNNs relative to the number of minority class samples, including the necessity to extract information invariant to lighting conditions, precipitation, or background colours from passing vehicles.

Classifications beyond binary (pin in or out of position) remain unexplored, such as scenarios where the pin ROI is wholly obscured or metal pins are partially out. Extending the binary classification to a multi-class system could allow future systems to detect various types of damage requiring different maintenance actions. Additional datasets capturing a broader range of anomalies, and more synthetic data would be required to support such enhancements, following the methodologies outlined in our synthetic data creation algorithm in Table 2. The average detection accuracy achieved was 0.93, which is commendable given the numerous challenges encountered during model training. Compared to other region-based detectors, our hybrid model offered higher accuracy and superior processing speed, handling 40 to 45 frames per second with up to 93.6% accuracy. The integration and depth concatenation layers enhanced the detection of smaller objects by incorporating low-level image details into the detection process, facilitated by a sequence of convolution, ReLU, and batch normalisation layers. The MATLAB app provides a robust platform for expanding research into future applications. In terms of 'dealing with the unknown' and research uncertainty, for the research community undertaking similar projects exceeding one or few years, it is worth considering additional challenges that are hard to predict.

Such considerations may include changes in industry partner staff, possible pandemic lockdowns, government funding, and policy updates, which require flexibility in project and data collection planning.

A summary of the practical aspects of this study are listed as follows:

- The technology offers a cost-effective automated solution to lane and general traffic safety, augmenting but not replacing human inspections.
- The system increases inspection frequency, enhances privacy, and enables the creation of digital records for analytical insights into traffic safety.
- Future scientific efforts will use a more extensive dataset to focus on adaptive model development and performance enhancement. Additional data visualisation and hybrid methodological approaches will be explored.
- For our industry partner, the NZTA, the project paves the way for independent software development and potential system integration into broader smart city infrastructures.
- The transition from a minimum viable product (MVP) to production systems will involve extensive testing, code optimisation, and, potentially, transitioning from MATLAB to Python to enhance computational efficiency integration capabilities and minimise the possibility of vendor locking into proprietary infrastructures and data processing outside of the national jurisdiction.
- This study lays the groundwork for future innovations in traffic management technology, positioning the NZTA to leverage the advancements in its ongoing modernisation of traffic infrastructure and smart city initiatives.

Future system advancements will consider enabling pin status tracking from various points of view, potentially including additional data collection protocols and technology, expanding the system's versatility and application scope.

5. Conclusions and Future Work

The Auckland Harbour Bridge plays a crucial role in Auckland's infrastructure, with traffic flows that are uneven but predictable, reversing in volume during morning and evening rush hours. Movable Concrete Barriers (MCBs) have proven effective in managing short-distance traffic bottlenecks; however, the bridge's susceptibility to various types of vibrations, particularly around its elevated central part, raises safety concerns, necessitating frequent pin inspections. Other health and safety concerns include sole reliance on manual inspections and the potential for human errors linked to inspection staff workplace safety, unhealthy spine ergonomic posture, and issues with protective gear.

To increase the safety and safety-monitoring frequency of the MCB, we developed a privacy-preserving automated monitoring system that is transferrable from data collected on the Auckland Harbour Bridge to similar contexts involving traffic flow regulation and safety monitoring applications [10]. A novel technique for generating synthetic frames was introduced to simulate various unsafe pin positions, aiding incremental model development and performance tuning. This research successfully demonstrated that the prototype can detect unsafe pin positions directly from live feeds and previously recorded video frames under varying lighting conditions (such as bright sunshine, heavy rain, and early morning 'soft' light conditions). The scarcity of video frames showing a Pin_Out status was addressed by introducing a method for creating synthetic images to enhance the modelling process. The system's expected overall performance for pin region detection, frame selection, and pin classification was anticipated to be above 80%, with individual models achieving up to 99% accuracy on a limited dataset, as shown in Table 5. These findings warrant further validation on a larger and more balanced future dataset. The pin status detection and alert system exhibited desirable precision and accuracy, with some performance decline attributed to the dataset's unbalanced nature, diverse lighting conditions, and camera angles and distance variations. Evidence from a smaller labelled dataset suggests that the system is a viable product that does not require further intensive

manual labelling. Integrating a hybrid model facilitated the analysis and provided flexibility for future model adjustments with minimal data labelling requirements.

Future work will include further video data collection, including additional videos recorded by the NZTA and AHB maintenance teams. The enhanced data collection is expected to bolster the foundational system and help further develop universally applicable ARDAD systems for similar traffic safety contexts globally. While the pin status detection and classification results are promising, there is significant potential for further advancements in integrating pin ROI tracking with the alert system. Future iterations of the system may also leverage advanced technologies such as LiDAR and GPS, which are becoming increasingly common in modern mobile devices. Developing additional capabilities will involve extensive system training and adaptation on enriched datasets that capture various pin conditions and scenarios, potentially leading to more robust and responsive traffic management solutions.

Author Contributions: Conceptualization, M.R.; Software, M.R.; Validation, M.R.; Data curation, M.R.; Writing—original draft, M.R.; Writing—review and editing, B.B. and M.D.; Supervision, B.B. and M.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Dataset available on request from the authors.

Acknowledgments: We extend our heartfelt thanks to Gary Bonser, Martin Olive, David Ranby, and Angela Potae from the NZTA and Auckland System Management (ASM) for their invaluable assistance with transport, safety briefings, video recordings, project insights, and ongoing support. We are also grateful to the Auckland University of Technology (AUT) for funding and PhD stipend support and for providing access to the computing hardware, library, and recording equipment. We also wish to acknowledge the documentation and software development efforts from industry, academia, and open-source communities producing MATLAB, Orange Data-Mining, SqueezeNet, TensorFlow, ImageNet, Google Cloud, and OpenCV.

Conflicts of Interest: The authors declare no conflict of interest.

Notations

The following table summarises the important symbols and mathematical notations used in this paper:

Symbol	Description
Y	Luminance component in the YCbCr colour space
x_c	Centroid location of an object in template matching
x_i	Pixel location in template matching
w_i	Pixel intensity in template matching
$P(X_t)$	Probability of observing a pixel value X at time t in a Gaussian Mixture Model
$w_{k,t}$	Weight of the k -th Gaussian component at time t
$\mu_{k,t}$	Mean of the k -th Gaussian at time t
$\Sigma_{k,t}$	Covariance matrix of the k -th Gaussian at time t
ΔMt	Motion detection metric combining frame differences and optical flow
F_t	Frame at time t
∇I_t	Gradient of the image at time t
v_t	Optical flow vector at time t
α	Weighting factor in motion detection equation
K_k	Kalman gain at time k
z_k	Actual measurement at time k
H_k	Measurement matrix that maps the state space into the measurement space

References

1. New Zealand Transport Agency Waka Kotahi. Auckland Harbour Bridge Factsheet. 2024. Available online: <https://www.nzta.govt.nz/assets/site-resources/content/about/docs/auckland-harbour-bridge-factsheet.pdf> (accessed on 10 April 2024).
2. Te Waihangā New Zealand Infrastructure Commission. New Zealand's Infrastructure Asset Value, Investment, and Depreciation, 1990–2022. 2023. Available online: <https://tewaihangā.govt.nz/our-work/research-insights/build-or-maintain> (accessed on 10 April 2024).
3. New Zealand Transport Agency Waka Kotahi. How to Move a Concrete Motorway Barrier. 2024. Available online: <https://www.nzta.govt.nz/media-releases/how-to-move-a-concrete-motorway-barrier/> (accessed on 10 April 2024).
4. Yang, X.; Zhang, J.; Liu, W.; Jing, J.; Zheng, H.; Xu, W. Automation in road distress detection, diagnosis and treatment. *J. Road Eng.* **2024**, *4*, 1–26. [\[CrossRef\]](#)
5. Bai, D.; Li, G.; Jiang, D.; Yun, J.; Tao, B.; Jiang, G.; Sun, Y.; Ju, Z. Surface defect detection methods for industrial products with imbalanced samples: A review of progress in the 2020s. *Eng. Appl. Artif. Intell.* **2024**, *130*, 107697. [\[CrossRef\]](#)
6. Trilles, S.; Hammad, S.S.; Iskandaryan, D. Anomaly detection based on artificial intelligence of things: A systematic literature mapping. *Internet Things* **2024**, *25*, 101063. [\[CrossRef\]](#)
7. Bačić, B.; Rathee, M.; Pears, R. Automating inspection of moveable lane barrier for Auckland harbour bridge traffic safety. In Proceedings of the Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, 23–27 November 2020; Proceedings, Part I 27. Springer: Berlin/Heidelberg, Germany, 2020; pp. 150–161.
8. Klarák, J.; Andok, R.; Malík, P.; Kuric, I.; Ritomský, M.; Klačková, I.; Tsai, H.-Y. From anomaly detection to defect classification. *Sensors* **2024**, *24*, 429. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Lozano-Ramírez, N.E.; Sánchez, O.; Carrasco-Beltrán, D.; Vidal-Méndez, S.; Castañeda, K. Digitalization and sustainability in linear projects trends: A bibliometric analysis. *Sustainability* **2023**, *15*, 15962. [\[CrossRef\]](#)
10. Wikipedia. Barrier Transfer Machine. 2024. Available online: https://en.wikipedia.org/wiki/Barrier_transfer_machine (accessed on 20 April 2024).
11. Rathee, M. *Safety Screening of Auckland's Harbour Bridge Movable Concrete Barrier*; Auckland University of Technology: Auckland, New Zealand, 2021.
12. Baccari, S.; Haddad, M.; Ghazzai, H.; Touati, H.; Elhadeif, M. Anomaly detection in connected and autonomous vehicles: A survey, analysis, and research challenges. *IEEE Access* **2024**, *12*, 19250–19276. [\[CrossRef\]](#)
13. Rathee, M.; Bačić, B.; Doborjeh, M. Automated road defect and anomaly detection for traffic safety: A systematic review. *Sensors* **2023**, *23*, 5656. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Cui, Y.; Liu, Z.; Lian, S. A survey on unsupervised anomaly detection algorithms for industrial images. *IEEE Access* **2023**, *11*, 55297–55315. [\[CrossRef\]](#)
15. Cottrell, B.H. Evaluation of a Movable Concrete Barrier System. [Tech Report] 1994. Available online: <https://rosap.ntl.bts.gov/view/dot/19352> (accessed on 9 April 2024).
16. Poe, C.M. Movable concrete barrier approach to the design and operation of a contraflow HOV lane. *Transp. Res. Rec.* **1991**, 40–54.
17. Chirayil Nandakumar, S.; Mitchell, D.; Erden, M.S.; Flynn, D.; Lim, T. Anomaly detection methods in autonomous robotic missions. *Sensors* **2024**, *24*, 1330. [\[CrossRef\]](#)
18. Galvão, Y.M.; Castro, L.; Ferreira, J.; Neto, F.B.d.L.; Fagundes, R.A.d.A.; Fernandes, B.J. Anomaly detection in smart houses for healthcare: Recent advances, and future perspectives. *SN Comput. Sci.* **2024**, *5*, 136. [\[CrossRef\]](#)
19. Es-Swidi, A.; Ardchir, S.; Elghoumari, Y.; Daif, A.; Azouazi, M. Traffic congestion and road anomalies detection using CCTVs images processing, challenges and opportunities. In *International Conference on Advanced Intelligent Systems for Sustainable Development*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 92–105.
20. Gao, J.; Zuo, F.; Ozbay, K.; Hammami, O.; Barlas, M.L. A new curb lane monitoring and illegal parking impact estimation approach based on queueing theory and computer vision for cameras with low resolution and low frame rate. *Transp. Res. Part A Policy Pract.* **2022**, *162*, 137–154. [\[CrossRef\]](#)
21. Kim, S.; Anagnostopoulos, G.; Barmounakis, E.; Geroliminis, N. Visual extensions and anomaly detection in the pNEUMA experiment with a swarm of drones. *Transp. Res. Part C Emerg. Technol.* **2023**, *147*, 103966. [\[CrossRef\]](#)
22. Yi, K.; Luo, K.; Chen, T.; Hu, R. An improved YOLOX model and domain transfer strategy for nighttime pedestrian and vehicle detection. *Appl. Sci.* **2022**, *12*, 12476. [\[CrossRef\]](#)
23. Wang, J.; Wang, X.; Hao, R.; Yin, H.; Huang, B.; Xu, X.; Liu, J. Incremental template neighborhood matching for 3D anomaly detection. *Neurocomputing* **2024**, *581*, 127483. [\[CrossRef\]](#)
24. Pan, Q.; Bao, Y.; Li, H. Transfer learning-based data anomaly detection for structural health monitoring. *Struct. Health Monit.* **2023**, *22*, 3077–3091. [\[CrossRef\]](#)
25. Yan, P.; Abdulkadir, A.; Luley, P.-P.; Rosenthal, M.; Schatte, G.A.; Grewe, B.F.; Stadelmann, T. A comprehensive survey of deep transfer learning for anomaly detection in industrial time series: Methods, applications, and directions. *IEEE Access* **2024**, *12*, 3768–3789. [\[CrossRef\]](#)
26. Bharambe, U.; Bhangale, U.; Narvekar, C. Role of multi-objective optimization in image segmentation and classification. In *Computational Intelligence in Image and Video Processing*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2023; pp. 317–340.
27. Gonzalez, R.C. *Digital Image Processing*; Pearson Education India: Hoboken, NJ, USA, 2009.

28. Yuan, Y.; Huang, J.; Yu, J.; Tan, J.K.S.; Chng, K.Z.; Lee, J.; Kim, S. Application of machine learning algorithms for accurate determination of bilirubin level on in vitro engineered tissue phantom images. *Sci. Rep.* **2024**, *14*, 5952. [\[CrossRef\]](#) [\[PubMed\]](#)
29. Kilicaslan, M.; Tanyeri, U.; Demirci, R. Image retrieval using one-dimensional color histogram created with entropy. *Adv. Electr. Comput. Eng.* **2020**, *20*, 79–88. [\[CrossRef\]](#)
30. Mittal, H.; Pandey, A.C.; Saraswat, M.; Kumar, S.; Pal, R.; Modwel, G. A comprehensive survey of image segmentation: Clustering methods, performance parameters, and benchmark datasets. *Multimed. Tools Appl.* **2022**, *81*, 35001–35026. [\[CrossRef\]](#)
31. Vansh, V.; Chandrasekhar, K.; Anil, C.; Sahu, S.S. Improved face detection using YCbCr and Adaboost. In Proceedings of the 5th International Conference on Computational Intelligence in Data Mining (ICCIDM 2018), Burla, India, 15–16 December 2018; Springer: Berlin/Heidelberg, Germany, 2020; pp. 689–699.
32. Han, H.; Han, C.; Lan, T.; Huang, L.; Hu, C.; Xue, X. Automatic shadow detection for multispectral satellite remote sensing images in invariant color spaces. *Appl. Sci.* **2020**, *10*, 6467. [\[CrossRef\]](#)
33. Sahu, Y.; Tripathi, A.; Gupta, R.K.; Gautam, P.; Pateriya, R.K.; Gupta, A. A CNN-SVM based computer aided diagnosis of breast Cancer using histogram K-means segmentation technique. *Multimed. Tools Appl.* **2023**, *82*, 14055–14075. [\[CrossRef\]](#)
34. Kollem, S.; Reddy, K.R.; Rao, D.S. An optimized SVM based possibilistic fuzzy c-means clustering algorithm for tumor segmentation. *Multimed. Tools Appl.* **2021**, *80*, 409–437. [\[CrossRef\]](#)
35. Agrawal, S.; Natu, P. An improved Gaussian Mixture Method based background subtraction model for moving object detection in outdoor scene. In Proceedings of the 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 15–17 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–8.
36. Rakesh, S.; Hegde, N.P.; Gopalachari, M.V.; Jayaram, D.; Madhu, B.; Hameed, M.A.; Vankdothu, R.; Kumar, L.S. Moving object detection using modified GMM based background subtraction. *Meas. Sens.* **2023**, *30*, 100898. [\[CrossRef\]](#)
37. Zhou, Q.; Situ, Z.; Teng, S.; Chen, G. Comparative Effectiveness of Data Augmentation Using Traditional Approaches versus StyleGANs in Automated Sewer Defect Detection. *J. Water Resour. Plan. Manag.* **2023**, *149*, 04023045. [\[CrossRef\]](#)
38. Zuehlke, D.; Henderson, T.A.; McMullen, S. Machine learning using template matching applied to object tracking in video data. *Artif. Intell. Mach. Learn. Multi-Domain Oper. Appl.* **2019**, *11006*, 110061S.
39. Ge, Y.; Zhang, J.; Ren, X.; Zhao, C.; Yang, J.; Basu, A. Deep variation transformation network for foreground detection. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 3544–3558. [\[CrossRef\]](#)
40. Cao, Q.; Wang, Z.; Long, K. Traffic foreground detection at complex urban intersections using a novel background dictionary learning model. *J. Adv. Transp.* **2021**, *2021*, 3515512. [\[CrossRef\]](#)
41. Zhang, Y.; Zheng, W.; Leng, K.; Li, H. Background subtraction using an adaptive local median texture feature in illumination changes urban traffic scenes. *IEEE Access* **2020**, *8*, 130367–130378. [\[CrossRef\]](#)
42. Feng, J.; Zeng, D.; Jia, X.; Zhang, X.; Li, J.; Liang, Y.; Jiao, L. Cross-frame keypoint-based and spatial motion information-guided networks for moving vehicle detection and tracking in satellite videos. *ISPRS J. Photogramm. Remote Sens.* **2021**, *177*, 116–130. [\[CrossRef\]](#)
43. Yang, G.; Ramanan, D. Upgrading optical flow to 3d scene flow through optical expansion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, Seattle, WA, USA, 13–19 June 2020; pp. 1334–1343.
44. Chen, X.; Jia, Y.; Tong, X.; Li, Z. Research on pedestrian detection and deepsort tracking in front of intelligent vehicle based on deep learning. *Sustainability* **2022**, *14*, 9281. [\[CrossRef\]](#)
45. Sun, M.; Davies, M.E.; Proudler, I.K.; Hopgood, J.R. Adaptive kernel Kalman filter. *IEEE Trans. Signal Process.* **2023**, *71*, 713–726. [\[CrossRef\]](#)
46. Chaurasiya, R.K.; Gondane, P.M.; Acharya, B.; Khan, M.I. Automatic road traffic analyzer using background subtraction, blob analysis, and tracking algorithms. In Proceedings of the 2023 7th International Conference on Computer Applications in Electrical Engineering-Recent Advances (CERA), Roorkee, India, 27–29 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
47. Sajeed, M.A.; Kelouwani, S.; Amamou, A.; Alam, M.Z.; Agbossou, K. Vehicle lane departure estimation on urban roads using GIS information. In Proceedings of the 2021 IEEE Vehicle Power and Propulsion Conference (VPPC), Gijon, Spain, 25–28 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–7.
48. Momin, K.A.; Barua, S.; Jamil, M.S.; Hamim, O.F. Short duration traffic flow prediction using kalman filtering. *AIP Conf. Proc.* **2023**, *2713*, 040011.
49. Moghaddasi, S.S.; Faraji, N. A hybrid algorithm based on particle filter and genetic algorithm for target tracking. *Expert Syst. Appl.* **2020**, *147*, 113188. [\[CrossRef\]](#)
50. Chen, S.; Huang, L.; Chen, H.; Bai, J. Multi-lane detection and tracking using temporal-spatial model and particle filtering. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 2227–2245. [\[CrossRef\]](#)
51. Nissimagoudar, P.; Algur, N.; Bonageri, N.; Chavan, A.; Koppa, A.; Iyer, N.C. Multiple vehicle tracking using meanshift algorithm and 8-point connectivity. In Proceedings of the International Conference on Soft Computing and Pattern Recognition, Online, 14–16 December 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 3–13.
52. Zhang, B.; Li, Z.; Perina, A.; Del Bue, A.; Murino, V.; Liu, J. Adaptive local movement modeling for robust object tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *27*, 1515–1526. [\[CrossRef\]](#)
53. O'Mahony, N.; Campbell, S.; Carvalho, A.; Harapanahalli, S.; Hernandez, G.V.; Krpalkova, L.; Riordan, D.; Walsh, J. Deep learning vs. traditional computer vision. In Proceedings of the Advances in Computer Vision: 2019 Computer Vision Conference (CVC), Las Vegas, NV, USA, 2–3 May 2019; Springer: Berlin/Heidelberg, Germany, 2020; Volume 1, pp. 128–144.

54. Zhang, W.; Li, H.; Li, Y.; Liu, H.; Chen, Y.; Ding, X. Application of deep learning algorithms in geotechnical engineering: A short critical review. *Artif. Intell. Rev.* **2021**, *54*, 5633–5673. [CrossRef]
55. Long, G.; Zhang, Z. Deep encrypted traffic detection: An anomaly detection framework for encryption traffic based on parallel automatic feature extraction. *Comput. Intell. Neurosci.* **2023**, 3316642. [CrossRef] [PubMed]
56. Butt, U.M.; Ullah, H.A.; Letchmunan, S.; Tariq, I.; Hassan, F.H.; Koh, T.W. Leveraging transfer learning for spatio-temporal human activity recognition from video sequences. *Comput. Mater. Contin.* **2023**, *74*, 5017–5033.
57. Ouhami, M.; Hafiane, A.; Es-Saady, Y.; El Hajji, M.; Canals, R. Computer vision, IoT and data fusion for crop disease detection using machine learning: A survey and ongoing research. *Remote Sens.* **2021**, *13*, 2486. [CrossRef]
58. Carlson, M.P.; Bloom, I. The cyclic nature of problem solving: An emergent multidimensional problem-solving framework. *Educ. Stud. Math.* **2005**, *58*, 45–75. [CrossRef]
59. Vallenga, D.; Grypdonck, M.H.F.; Hoogwerf, L.J.R.; Tan, F.I.Y. Action research: What, why and how? *Acta Neurol. Belg.* **2009**, *109*, 81–90. [PubMed]
60. Farady, I.; Lin, C.-Y.; Chang, M.-C. PreAugNet: Improve data augmentation for industrial defect classification with small-scale training data. *J. Intell. Manuf.* **2024**, *35*, 1233–1246. [CrossRef]
61. Xu, M.; Yoon, S.; Fuentes, A.; Park, D.S. A comprehensive survey of image augmentation techniques for deep learning. *Pattern Recognit.* **2023**, *137*, 109347. [CrossRef]
62. Mathworks. Develop Apps Using App Designer. 2021. Available online: <https://www.mathworks.com/help/matlab/app-designer.html> (accessed on 10 April 2024).
63. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
64. Sutskever, I.; Martens, J.; Dahl, G.; Hinton, G. On the importance of initialization and momentum in deep learning. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; PMLR; pp. 1139–1147.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Comment

Comment on Sakli, H. Cylindrical Waveguide on Ferrite Substrate Controlled by Externally Applied Magnetic Field. *Electronics* 2021, 10, 474

Afshin Moradi 

Department of Engineering Physics, Kermanshah University of Technology, Kermanshah 6715685420, Iran; a.moradi@kut.ac.ir

Abstract: Recently, Sakli investigated the propagation of electromagnetic waves in metallic cylindrical waveguides filled with longitudinally magnetized ferrite, focusing on TE_z (transverse electric) and TM_z (transverse magnetic) modes relative to the z -axis. This commentary highlights that the proposed system generally cannot support the propagation of the TE_z and TM_z waves, rendering the main results derived by Sakli invalid.

Keywords: anisotropic materials; antenna; cylindrical waveguides; ferrites; propagation

1. Introduction

Recently, Sakli [1] investigated the propagation of TE_z and TM_z modes in metallic cylindrical waveguides filled with lossless, longitudinally magnetized ferrite. He demonstrated how to obtain dispersion diagrams and discussed the impact of anisotropic parameters on dispersion characteristics and cutoff frequencies. Additionally, he presented numerical results for the TE_z and TM_z modes.

However, based on the theory of ferrite-filled cylindrical waveguides obtained in the beginning of the 50s of the last century [2,3], the hybrid wave should be expected for the proposed metallic cylindrical waveguide propagation, and therefore TE_z and TM_z waves are unable to propagate. Additionally, as shown in our recent study [4], a metallic cylindrical waveguide filled with homogeneous anisotropic materials generally supports only hybrid modes. This means that the above-mentioned results for the propagation of TE_z and TM_z modes derived by Sakli [1] are invalid. Let us note that it was also well established that in general cases, the separation of electromagnetic waves into TE and TM modes is not possible in metallic rectangular waveguides [5,6].

We believe that incorrect publications should be corrected to ensure new researchers can build upon accurate prior studies. This motivation drives our commentary, in which we identify the errors in [1].

2. The Rigorous Electromagnetic Analysis

Consider a metallic cylindrical waveguide filled with longitudinally magnetized ferrite as shown in Figure 1 of [1]. We are interested in guided mode solutions in the waveguide propagating in the z -direction. Therefore, let us consider the form of the electromagnetic wave propagating in the waveguide as

$$\mathbf{E}(r, \theta, z, t) = \mathbf{E}(r, \theta)e^{j(\omega t - k_z z)}, \quad (1)$$

$$\mathbf{H}(r, \theta, z, t) = \mathbf{H}(r, \theta)e^{j(\omega t - k_z z)}, \quad (2)$$



Citation: Moradi, A. Comment on Sakli, H. Cylindrical Waveguide on Ferrite Substrate Controlled by Externally Applied Magnetic Field. *Electronics* 2021, 10, 474. *Electronics* **2024**, *13*, 3031. <https://doi.org/10.3390/electronics13153031>

Academic Editor: Flavio Canavero

Received: 9 May 2024

Revised: 11 July 2024

Accepted: 29 July 2024

Published: 1 August 2024



Copyright: © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

where k_z is the propagation constant along the z -direction. Expanding the Maxwell curl equations (i.e., Equations (1) and (2) in [1]) and using Equation (3) in [1] we obtain

$$\begin{pmatrix} \frac{1}{r} \frac{\partial E_z}{\partial \theta} - \frac{\partial E_\theta}{\partial z} \\ \frac{\partial E_r}{\partial z} - \frac{\partial E_z}{\partial r} \\ \frac{1}{r} \frac{\partial(rE_\theta)}{\partial r} - \frac{1}{r} \frac{\partial E_r}{\partial \theta} \end{pmatrix} = -j\omega\mu_0 \begin{pmatrix} \mu H_r - j\kappa H_\theta \\ j\kappa H_r + \mu H_\theta \\ \mu_{rz} H_z \end{pmatrix}, \quad (3)$$

$$\begin{pmatrix} \frac{1}{r} \frac{\partial H_z}{\partial \theta} - \frac{\partial H_\theta}{\partial z} \\ \frac{\partial H_r}{\partial z} - \frac{\partial H_z}{\partial r} \\ \frac{1}{r} \frac{\partial(rH_\theta)}{\partial r} - \frac{1}{r} \frac{\partial H_r}{\partial \theta} \end{pmatrix} = j\omega\epsilon_0\epsilon_{rf} \begin{pmatrix} E_r \\ E_\theta \\ E_z \end{pmatrix}. \quad (4)$$

Putting Equations (1) and (2) in Equations (3) and (4), and conducting some manipulation, we obtain

$$E_r = \frac{k_z}{K_c^2} \left(-jK_{c\mu}^2 \frac{\partial E_z}{\partial r} + F \frac{1}{r} \frac{\partial E_z}{\partial \theta} \right) - \frac{1}{K_c^2} \left(A_1 \frac{\partial H_z}{\partial r} + jA_2 \frac{1}{r} \frac{\partial H_z}{\partial \theta} \right), \quad (5)$$

$$E_\theta = -\frac{k_z}{K_c^2} \left(F \frac{\partial E_z}{\partial r} + jK_{c\mu}^2 \frac{1}{r} \frac{\partial E_z}{\partial \theta} \right) + \frac{1}{K_c^2} \left(jA_2 \frac{\partial H_z}{\partial r} - A_1 \frac{1}{r} \frac{\partial H_z}{\partial \theta} \right), \quad (6)$$

$$H_r = \frac{\omega\epsilon_0\epsilon_{rf}}{K_c^2} \left(F \frac{\partial E_z}{\partial r} + jK_{c\mu}^2 \frac{1}{r} \frac{\partial E_z}{\partial \theta} \right) - \frac{1}{K_c^2} \left(jk_z K_{c\mu}^2 \frac{\partial H_z}{\partial r} - Fk_z \frac{1}{r} \frac{\partial H_z}{\partial \theta} \right), \quad (7)$$

$$H_\theta = \frac{\omega\epsilon_0\epsilon_{rf}}{K_c^2} \left(-jK_{c\mu}^2 \frac{\partial E_z}{\partial r} + F \frac{1}{r} \frac{\partial E_z}{\partial \theta} \right) - \frac{1}{K_c^2} \left(Fk_z \frac{\partial H_z}{\partial r} + jk_z K_{c\mu}^2 \frac{1}{r} \frac{\partial H_z}{\partial \theta} \right), \quad (8)$$

where some symbols are introduced for convenience, namely the following:

$$\begin{aligned} K_{c\mu}^2 &= \epsilon_{rf} k_0^2 \mu - k_z^2, \\ F &= \epsilon_{rf} k_0^2 \kappa, \\ K_c^2 &= K_{c\mu}^4 - F^2, \\ A_1 &= \frac{Fk_z^2}{\omega\epsilon_0\epsilon_{rf}}, \\ A_2 &= \frac{K_c^2 + k_z^2 K_{c\mu}^2}{\omega\epsilon_0\epsilon_{rf}}. \end{aligned} \quad (9)$$

We remind the reader that the remaining parameters in the equations used in this work were defined in reference [1]. Note that the longitudinal components satisfy the following coupled equations:

$$\left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2} + \frac{\omega\mu_0\mu_{rz}K_c^2}{A_2} \right] H_z = -j \frac{k_z F}{A_2} \left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2} \right] E_z, \quad (10)$$

$$\left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2} + \frac{K_c^2}{K_{c\mu}^2} \right] E_z = \frac{jFk_z}{K_{c\mu}^2 \omega\epsilon_0\epsilon_{rf}} \left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2} \right] H_z, \quad (11)$$

so that, in general, a hybrid mode is needed. Note that the right-hand sides of Equations (10) and (11) were neglected in the analysis by Sakli [1]; therefore, his statement that TE_z and TM_z modes can be supported separately in a metallic cylindrical waveguide filled with longitudinally magnetized ferrite is incorrect.

However, the decoupling of E_z from H_z occurs in two particular cases. In the first case, Equations (10) and (11) can be separated into two independent equations of E_z and H_z , when the magnetization is equal to zero, i.e., when we have

$$\kappa = 0. \quad (12)$$

In this case, Equations (10) and (11) become

$$\left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2} + \frac{\mu_{rz}}{\mu} \left(\varepsilon_{rf} k_0^2 \mu - k_z^2 \right) \right] H_z = 0, \quad (13)$$

$$\left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2} + \varepsilon_{rf} k_0^2 \mu - k_z^2 \right] E_z = 0, \quad (14)$$

and therefore, TE_z and TM_z modes can be supported separately in a metallic cylindrical waveguide filled with longitudinally “unmagnetized” ferrite. In the second case, decoupling occurs if we consider the azimuthal modes, i.e., when we have

$$k_z = 0. \quad (15)$$

Putting the propagation constant equal to zero in Equations (10) and (11), we obtain

$$\left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2} + \varepsilon_{rf} \mu_{rz} k_0^2 \right] H_z = 0, \quad (16)$$

$$\left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2} + \frac{\varepsilon_{rf}}{\mu} \left(\mu^2 - \kappa^2 \right) k_0^2 \right] E_z = 0. \quad (17)$$

In this case, TE_z and TM_z modes with components (H_z, E_r, E_θ) and (E_z, H_r, H_θ) appear, respectively.

3. Conclusions

In this work, we began by using the general mathematical framework for how electromagnetic waves travel through a metallic cylindrical waveguide that is filled with a ferrite material magnetized along its length (as outlined in references [2] and [3]). Our analysis demonstrated that, in most situations, this type of system does not permit the distinct separation of TE_z and TM_z modes. There are only two special scenarios where this separation might occur. As a result, the main findings presented by Sakli in reference [1] are invalid.

Funding: This research received no external funding.

Data Availability Statement: The data that supports the findings of this study are available within the article.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Sakli, H. Cylindrical Waveguide on Ferrite Substrate Controlled by Externally Applied Magnetic Field. *Electronics* **2021**, *10*, 474. [\[CrossRef\]](#)
2. Kales, M.L. Modes in wave guides containing ferrites. *J. Appl. Phys.* **1953**, *24*, 604. [\[CrossRef\]](#)
3. Gamo, H. The Faraday rotation of waves in a circular waveguide. *J. Phys. Soc. Jpn.* **1953**, *8*, 176. [\[CrossRef\]](#)
4. Moradi, A.; Bait-Suwailam, M.M. Comment on: Enhanced coupling of light from subwavelength sources into a hyperbolic metamaterial fiber. *J. Light. Technol.* **2024**, *42*, 5435–5436. [\[CrossRef\]](#)
5. Moradi, A. Comment on controllable metamaterial loaded waveguides supporting backward and forward waves. *IEEE Trans. Antennas Propag.* **2023**, *72*, 3858–3859. [\[CrossRef\]](#)
6. Moradi, A.; Bait-Suwailam, M.M. Magnetostatic waves in metallic rectangular waveguides filled with uniaxial negative permeability media. *J. Appl. Phys.* **2024**, *135*, 153102. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Power Pylon Type Identification and Characteristic Parameter Calculation from Airborne LiDAR Data

Shengxuan Zu ^{1,2} , Linong Wang ^{1,2,*}, Shaocheng Wu ^{1,2}, Guanjian Wang ^{1,2}  and Bin Song ^{1,2}

¹ Engineering Research Center of Ministry of Education for Lightning Protection and Grounding Technology, School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China; 2022202070038@whu.edu.cn (S.Z.); wushaocheng@whu.edu.cn (S.W.); wangguanjian@whu.edu.cn (G.W.); 00007609@whu.edu.cn (B.S.)

² School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China

* Correspondence: wangln@whu.edu.cn; Tel.: +86-138-7120-8599

Abstract: Reconstructing three-dimensional (3D) models of power equipment plays an increasingly important role in advancing digital twin power grids. To reconstruct a high-precision model, it is crucial to accurately obtain the pylon type and its necessary parameter information before modeling. This study proposes an improved method for identifying pylon types based on similarity measurement and a linearly transformed dataset. It begins by simplifying the identification of point clouds using the pylon shape curve. Subsequently, the resemblance between the curve and those curves within the dataset is evaluated using a similarity measurement to determine the pylon type. A novel method is proposed for calculating the characteristic parameters of the pylon point clouds. The horizontal and vertical distribution characteristics of the pylon point clouds are analyzed to identify key segmentation positions based on their types. Feature points are derived from key segmentation positions to calculate the characteristic parameters. Finally, the pylon 3D models are reconstructed on the basis of the calculated values. The experimental results showed that, compared with other similarity measurements, the Hausdorff distance had the best effect as a similarity measurement using the linearly transformed dataset, with an overall evaluation F-score of 86.4%. The maximum relative error of the calculated pylon parameters did not exceed 5%, affirming the feasibility of the algorithm.

Keywords: airborne LiDAR; power pylon; similarity measurement; 3D reconstruction



Citation: Zu, S.; Wang, L.; Wu, S.; Wang, G.; Song, B. Power Pylon Type Identification and Characteristic Parameter Calculation from Airborne LiDAR Data. *Electronics* **2024**, *13*, 3032. <https://doi.org/10.3390/electronics13153032>

Academic Editor: Ahmed Abu-Siada

Received: 17 June 2024

Revised: 22 July 2024

Accepted: 30 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid advancement of the national economy, there is a continuous surge in demand for fundamental energy sources like electricity. High-voltage transmission lines, serving as essential infrastructure for long-distance power transmission, play a crucial role in national economic development and daily production [1–3]. Regular monitoring and maintenance of power transmission corridors are imperative to ensure the safe and stable operation of the power system. Traditional inspection of transmission lines is typically conducted by line inspectors who visually or with handheld instruments inspect power equipment and identify potential hazards based on experience. However, due to several constraints and the absence of three-dimensional (3D) data along the lines, manual inspections entail a significant workload, low precision, and potential safety hazards, rendering them inadequate to meet the inspection needs of the power grid [4–7].

In recent years, airborne Light Detection and Ranging (LiDAR) technology has been widely used in power inspections because it enables the rapid and precise acquisition of dense 3D point clouds within power transmission corridors without the limitations of light and terrain [8–10]. As an essential component of transmission lines, the power pylon plays a pivotal role in ensuring the safety of high-voltage lines. The reconstructed 3D model of the pylon can provide basic data and model support for conducting multi-physics field simulation analysis, simulating real working scenes, and guiding the selection of

operational methods. Therefore, it is necessary to develop an accurate and efficient method to extract pylon information from point clouds for 3D model reconstruction.

Currently, the research on pylon point clouds primarily focuses on the segmentation of pylon points from the point clouds collected by airborne LiDAR [11–16]. And there are many methods for pylon reconstruction based on point cloud data, which can be classified as data-driven [17], model-driven [10], and hybrid-driven [18]. The data-driven approach is generally a bottom-up strategy. It directly processes the data without the need to presuppose the characteristics of the reconstructed object. Han et al. [17] proposed a data-driven method of modeling the power pylon. In this method, power pylon point clouds were located and extracted by utilizing the connection points of line pairs. The 3D model of the power pylon was constructed according to the 3D line feature obtained from the binary image contour tracking. This method requires high data quality, and it is difficult to reconstruct the pylon structure when there are many noise points in the obtained point clouds. Compared with the data-driven approach, the model-driven approach takes a top-down strategy and requires a model library to be completed in advance. Li et al. [10] divided the pylon into three relatively simple parts: the foot, the body, and the head. The head was reconstructed by seeking the corresponding model from the pre-built model library, and the body was reconstructed by calculating the intersection lines of the fitted side planes. The experiment suggested that the approach can achieve automatic 3D modeling of the pylon head and body effectively. However, the reconstruction of the pylon foot required interactive operation. Since it is difficult to meet the reconstruction requirements of complex objects by using data-driven or model-driven methods alone, hybrid methods combining the above two methods have been proposed. The method adopts appropriate strategies according to different structural modeling requirements, which can improve the modeling accuracy. Zhou et al. [18] divided the pylon into the head and body. They reconstructed the pylon body by a data-driven strategy and the head by a model-driven strategy with the aid of a predefined 3D head model library. This method can accurately reconstruct the original pylon structure, but it cannot effectively handle pylons containing more complex structures.

To solve the problems in the above modeling methods, the pylon types and necessary parameter information obtained from 3D point clouds are used to reconstruct the pylons in this paper.

The existing methods for identifying the pylon types are mainly classified into rule-based methods and machine learning methods [19]. For the first method, the characteristics are extracted from the pylon point clouds, and then the pylon types are identified according to the difference of the characteristics. Qiao et al. [20] layered the pylon head point clouds and then calculated the rate of vertical filling for every individual layer. They classified pylons into two types based on the position of the layer with the largest filling rate. Chen et al. [21] segmented the pylon head point clouds based on point distribution characteristics and then projected it onto the Y_0Z_0 plane to create an image. The pylon head contour image was acquired by integrating the image processing method. Finally, the pylon type was determined based on the quantity of pixels within the contour. Although the above two methods can distinguish pylons, the types of pylons that can be identified are very limited. Silva F. et al. [22] proposed a classification methodology based on similarity. They utilized point cloud distance metrics to measure the similarity between pylon point clouds and basic reference models, achieving pylon classification based on differences in distance. This method requires handling large amounts of data and is sensitive to fluctuations in the density of sampling points. For the second method, the identification of the pylon types is realized based on the machine learning algorithm. Zhou et al. [18] first defined a 3D parameterized model library of pylon heads. Then, pylon head types were identified by the shape context algorithm, and a simulated annealing algorithm was used to estimate the relevant parameters of the pylon heads. Chen et al. [23] extracted features based on point elevation histograms and frontal projection, and finally used the Support Vector Machine (SVM) classification method to train and classify head feature vector samples.

Wang et al. [24] projected inner and outer contour points into rasterized images to extract Histogram of Oriented Gradient (HOG) features. Then, they used these as inputs to the SVM classifier for type identification. However, when the SVM algorithm processes large-scale data sets, it may take longer to train due to its higher computational complexity and storage requirements.

In summary, the current methods for identifying pylon types suffer from limited recognition effectiveness, susceptibility to variations in sampling point density, high computational complexity, and long training times. Moreover, there is no suitable method for calculating the parameters of the pylon point clouds. To solve the aforementioned problems, this study proposes an improved method for identifying pylon types based on similarity measurement and a novel characteristic parameter calculation method of pylon point clouds. Figure 1 shows the data processing flow chart of this paper. Firstly, the point clouds are preprocessed with zero-mean normalization, shifting, and redirection before the pylon information is obtained. Secondly, pylon types are identified by calculating the similarity measurement between the shape curves generated based on the point clouds and curves within the dataset. Then, the pylon characteristic positions are determined based on the calculated point clouds' number, density, filling rate, and shape parameter. In the case of obtaining these positions, the feature points are derived from the point clouds to calculate the characteristic parameters. Finally, the pylon 3D models are reconstructed on the basis of the calculated values.

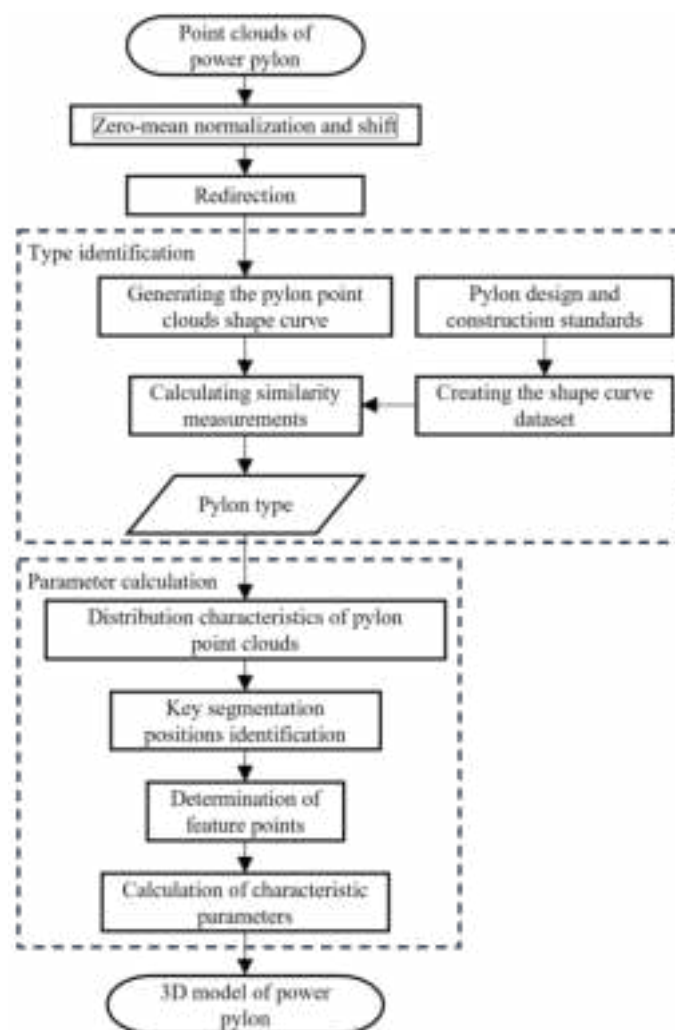


Figure 1. Overall flow chart of type identification and parameter calculation.

The main innovations and contributions of this study are as follows:

1. This study proposes an improved method for identifying pylon types based on the similarity measurement and a linearly transformed dataset. Comparing the effects of four similarity measurements on pylon type identification, a similarity measurement that is more suitable for various pylon type identification is obtained.
2. A novel method for automatic calculation of characteristic parameters using airborne LiDAR data is proposed for the first time. It can efficiently extract the specific information of the pylon, ensuring the high accuracy of characteristic parameters.
3. The 3D models of the four types of pylons are reconstructed on the basis of the identified pylon types and calculated parameters, which can accurately reflect the true structure of the pylons.

2. Pylon Point Cloud Type Identification Based on Similarity Measurements

In the process of collecting pylon point clouds using airborne LiDAR, the majority of laser points are not obtained by scanning the surface of the target vertically but rather through inclined incidence. Therefore, in addition to being located on the top of the pylon, some of the pylon point clouds are situated on the sides of the pylon. Additionally, in the absence of Unmanned Aerial Vehicle (UAV) flight routes and specific spatial geographic information on the pylons, it is impossible to determine which side the points belong to. Furthermore, the pylon structure is generally complex. If all the pylon point clouds are utilized to determine the pylon type in 3D space, the aforementioned circumstances will make it difficult to achieve this goal.

To address this issue more effectively, this paper introduces pylon shape curves, simplifying the pylon type identification in 3D space to the identification of curves in two-dimensional (2D) space. This curve can accurately reflect the shape of the pylon. The shapes of various pylon types exhibit significant differences, and the shapes of pylons that belong to the same type but different models are basically similar. Finally, the similarity measurements are combined to determine the resemblance between the shape curve derived from the pylon point clouds and the curves within the dataset, and then the type of the pylon is determined.

2.1. Point Cloud Preprocessing

Considering that the pylon point clouds collected by airborne LiDAR can be oriented arbitrarily in 3D space, for the sake of facilitating similarity measurement calculation and subsequent processing, we perform zero-mean normalization on the pylon point cloud data using Equation (1). Then, we transform the coordinates of the processed points by shifting them along the Z-axis positive direction, obtaining the updated coordinates (x' , y' , z').

$$\begin{cases} x' = x - x_0 \\ y' = y - y_0 \\ z' = z - z_0 - \min(z - z_0) \end{cases} \quad (1)$$

where x_0 , y_0 , and z_0 represent the coordinates of the central position of the initial point clouds; and x' , y' , and z' represent the coordinates of the point clouds after zero-mean normalization and shifting.

The pylon structure is generally symmetrical, and capturing the shape characteristics of the pylon from the front view is more effective. Consequently, it is required to rotate the point clouds of the pylon by a certain angle, θ , before generating the pylon shape curve, aligning its horizontal direction perpendicular to the X-axis.

The horizontal direction of the pylon is generally more relevant to its upper structure. In this study, point clouds with Z coordinates exceeding H are chosen and projected onto the XY plane. Subsequently, the eigenvalues and eigenvectors of the resulting projected point clouds are calculated using the principal component analysis (PCA) algorithm. The obtained minimum eigenvalue corresponds to the eigenvector (v_1 , v_2) perpendicular to the point cloud horizontal orientation. And the rotation angle θ is calculated by

Equation (2) [20]. Finally, the coordinate transformation is conducted using Equation (3) to obtain the coordinates (x'', y'', z'') of the rotated point clouds. Figure 2 displays the projections of the redirected pylons.

$$\theta = \arccos\left(\frac{v_1}{\sqrt{v_1^2 + v_2^2}}\right) \quad (2)$$

$$\begin{cases} x'' = x' \cos(\theta) - y' \sin(\theta) \\ y'' = x' \sin(\theta) + y' \cos(\theta) \\ z'' = z' \end{cases} \quad (3)$$

where $x'', y'',$ and z'' represent the coordinates of the rotated point clouds.

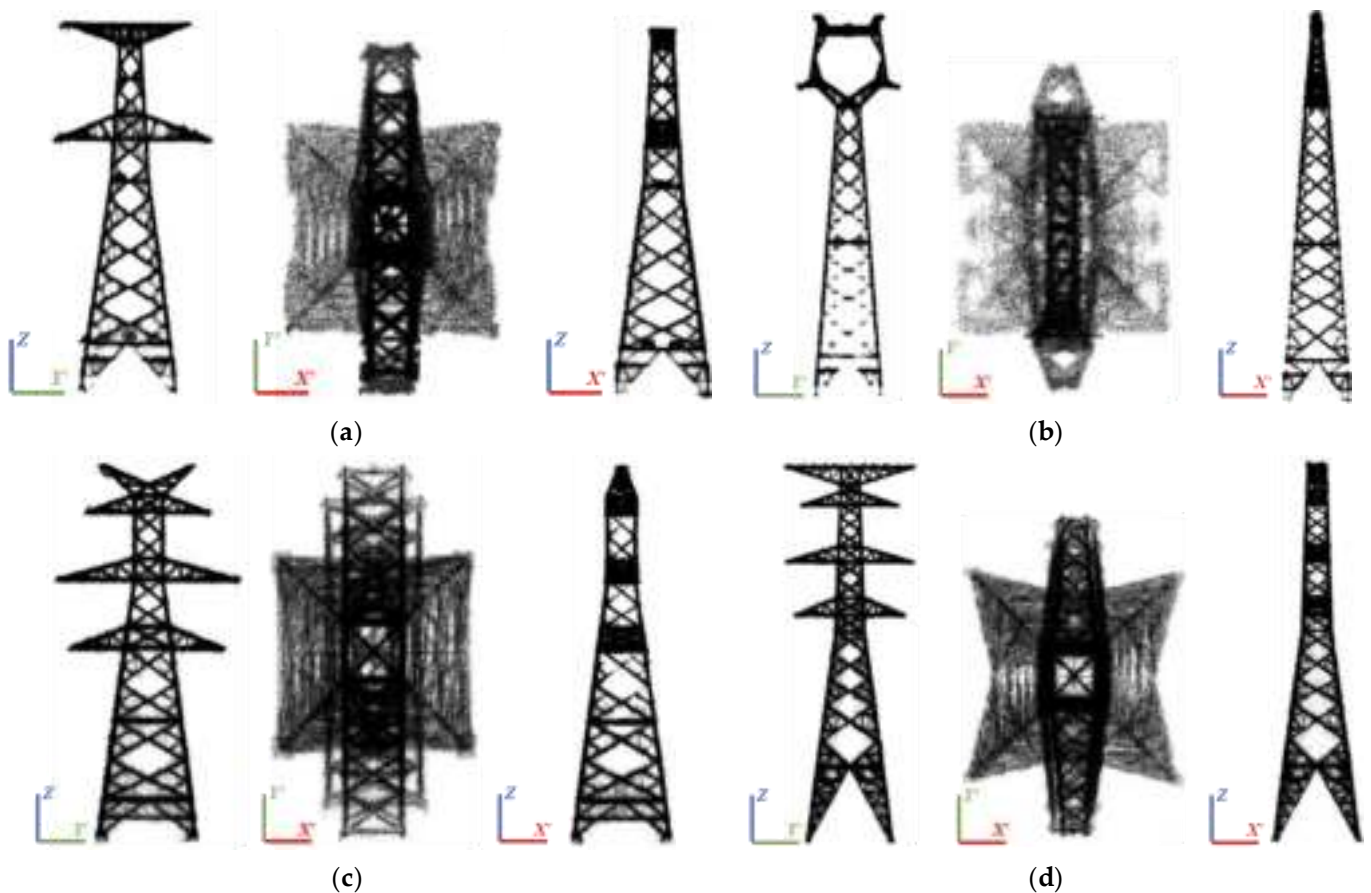


Figure 2. The projections of the redirected pylons on the $Y'Z$, $X'Y'$, and $X'Z$ planes. (a) Type-a pylon. (b) Type-b pylon. (c) Type-c pylon. (d) Type-d pylon.

2.2. Generating the Pylon Point Cloud Shape Curve

The redirected pylon point clouds are projected onto the $Y'Z$ plane, and then vertically layered along the Z -axis at a certain interval of h_1 . The boundary points of each layer are found by using a sliding window. These points form the overall pylon shape curve. Considering the symmetry of the pylon structure, half of the boundary points can be selected and connected in sequence. Finally, uniformly spaced discrete curves, which are the pylon point cloud shape curves, are obtained, as shown in Figure 3.

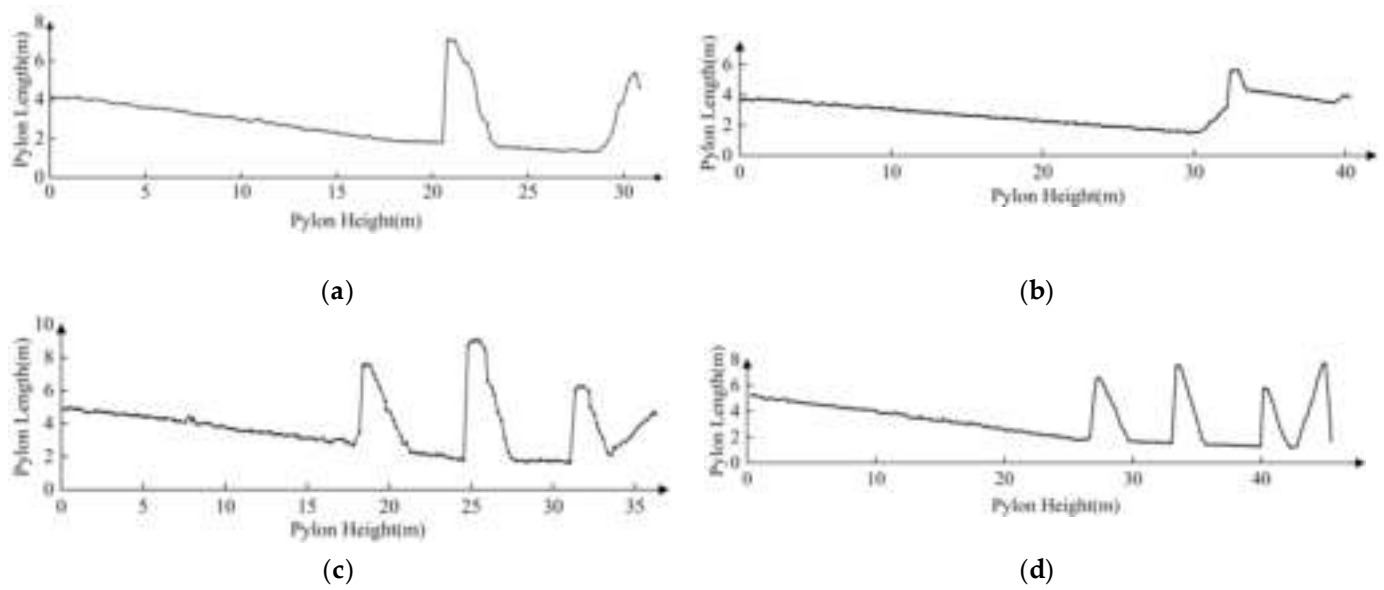


Figure 3. Pylon point cloud shape curve. (a) Type-a pylon. (b) Type-b pylon. (c) Type-c pylon. (d) Type-d pylon.

2.3. Creating the Shape Curve Dataset

The outer shape of pylons serves as a crucial criterion for identifying pylon types. Before calculating similarity measurements, it is essential to create a dataset of pylon shape curves. The creation of this dataset refers to relevant general design and typical design standards issued by the State Grid Corporation of China, along with other pylon design and construction standards, ensuring the accuracy and completeness of the data. Based on common pylon parameters provided by these standards, such as height, length, cross-arm length, and other information in the vertical and horizontal directions, the shape curves of different pylons in the dataset are generated. Then, the shape curves in the dataset are linearly transformed to the same height, as shown in Figure 4. And to facilitate the calculation of the similarity measurement, half of the feature points are connected to form a discrete curve before utilization.

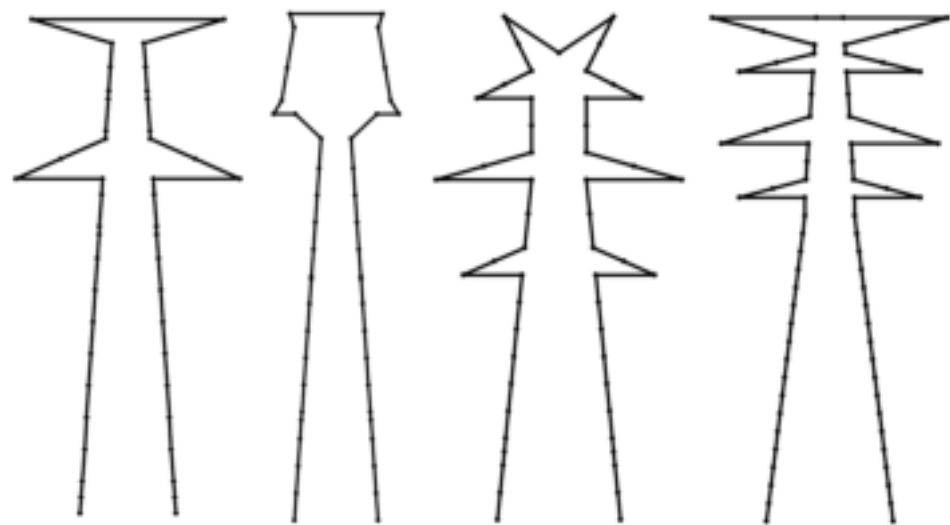


Figure 4. Shape curves of some pylons in the dataset.

Currently, the dataset contains the shape curves of various common power pylons, such as cat-head pylons, sheep horn pylons, and cup pylons. These curves are generated based on standard parameters. In addition, taking into account the correlation between the number of pylon arms and phases, and the orientation of the power lines, we classify these pylons into single-phase, two-phase, and three-phase pylons. Furthermore, we also consider the cases of single-circuit, double-circuit, and four-circuit pylons [25].

2.4. Similarity Measurements

The shape curve of a pylon can provide various information, such as the shape of the pylon head, pylon height, and the number of crossarms. Reasonably utilizing the distinguishing features of different types of pylons will contribute to accurately determining the pylon type. Moreover, the selection of similarity measurements is crucial for the identification of pylon types. This study selects four widely used similarity evaluation measurements in the fields of pattern recognition and artificial intelligence: the Dynamic Time Warping (DTW) distance, FastDTW distance, discrete Fréchet distance, and Hausdorff distance. We compare their effectiveness in identifying pylon types.

(1) DTW distance and FastDTW distance: The DTW algorithm is widely used in evaluating the resemblance of time sequences with different lengths, and has extensive applications in the field of speech recognition [26,27]. Its main approach is as follows.

Suppose there are two time sequences, $X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$ and $Y = \{y_1, y_2, \dots, y_j, \dots, y_m\}$. The cumulative distance matrix D of X and Y is constructed using Euclidean distance. A warping path W in matrix D is found such that the sum of elements along the path is minimized. The minimum cumulative distance can be calculated using Equation (4) by satisfying both the monotonicity and continuity constraints.

$$\begin{cases} D(i, j) = d(x_i, y_j) + \min\{D(i-1, j), D(i, j-1), D(i-1, j-1)\} \\ 1 \leq i \leq n, \quad 1 \leq j \leq m \end{cases} \quad (4)$$

where $d(x_i, y_j)$ represents the distance between x_i and y_j . $D(i, j)$ represents the cumulative distance between time steps x_i and y_j .

The final minimum cumulative distance can be used to assess the similarity of sequences X and Y . A smaller value indicates that the two sequences share a greater resemblance in terms of their shape; conversely, a larger value suggests less similarity.

However, when the time sequences are lengthy, the computational complexity of calculating the DTW distance between the two sequences is $O(nm)$, leading to relatively low algorithm efficiency. In this case, the DTW algorithm is usually accelerated by limiting the path search range, data abstraction, and indexing. FastDTW uses the first two methods to expedite DTW. This improved algorithm effectively decreases the time complexity of DTW to $O(m)$. FastDTW primarily involves three processes: coarsening, projection, and refinement [28–30].

(2) Discrete Fréchet distance: The discrete Fréchet distance takes into account the shape of curves as well as the sequence of points along the curves. It is a distance measure to determine the degree of similarity between curves and is employed in various fields to gauge the similarity between parameterized curves [31,32]. Its definition is as follows.

If there are two polygon curves X and Y consisting of m and n points, respectively. To calculate the discrete Fréchet distance between X and Y , the corresponding sequence of point pairs is found at first.

$$L = \{(x_{a_1}, y_{b_1}), (x_{a_2}, y_{b_2}), \dots, (x_{a_k}, y_{b_k})\} \quad (5)$$

Among them, $a_1 = 1, b_1 = 1, a_k = m$, and $b_k = n$. And to ensure the order of points, for any $i = 1, \dots, k$, there is $a_{i+1} = a_i$ or $a_{i+1} = a_i + 1, b_{i+1} = b_i$, or $b_{i+1} = b_i + 1$. Then, we calculate the maximum distance between corresponding point pairs.

$$\|L\| = \max_{i=1, \dots, k} d(x_{a_i}, y_{b_i}) \quad (6)$$

The discrete Fréchet distance between X and Y is defined as follows:

$$D_f(X, Y) = \min\{\|L\|\} \quad (7)$$

$D_f(X, Y)$ can be used to assess the similarity between X and Y . The resemblance of the shapes between both curves increases as the value decreases.

(3) Hausdorff distance: The Hausdorff distance describes the similarity of two subsets by measuring the distance between them in space [33]. If there are two finite point sets X and Y , with lengths n and m , respectively, then the bidirectional Hausdorff distance $D_h(X, Y)$ of these two sets of data is:

$$\begin{cases} D_h(X, Y) = \max\{d_h(X, Y), d_h(Y, X)\} \\ d_h(X, Y) = \max_{x \in X} \min_{y \in Y} \|x - y\| \\ d_h(Y, X) = \max_{y \in Y} \min_{x \in X} \|y - x\| \end{cases} \quad (8)$$

where $D_h(X, Y)$ takes the maximum value between $d_h(X, Y)$ and $d_h(Y, X)$. $\|\cdot\|$ represents the Euclidean distance between point sets X and Y . $d_h(X, Y)$ is the maximum shortest distance from the point in X to the Y set. $d_h(Y, X)$ represents the maximum shortest distance from the point in Y to the X set.

The Hausdorff distance measures the dissimilarity of two sets of points and can be used to assess the similarity of X and Y . A smaller value of $D_h(X, Y)$ indicates a greater similarity in shape between X and Y .

3. Characteristic Parameter Calculation of Pylon Point Clouds

In this section, the distribution characteristics of the pylon point clouds including number, density, and horizontal filling rate are first calculated. Then, these characteristics and the pylon type are used to identify the key segmentation positions. Finally, based on these positions, feature points are derived from the point clouds to calculate the characteristic parameters.

3.1. Distribution Characteristics of Pylon Point Clouds

The pylon point clouds projected onto the $Y'Z$ plane are vertically layered to generate histograms of the distribution characteristic value. Then, a sliding window is used to identify layers that simultaneously satisfy both the local maximum number of point clouds and the local maximum point cloud density. And the horizontal filling rate of these layers is calculated. Finally, key segmentation positions are identified from the layers with the great filling rate.

The point clouds are layered along the Z -axis with a fixed interval h_1 . The number of pylon points and the spatial size of each layer are calculated and used to obtain the distribution degree of the point clouds in each layer, that is, the point cloud density. In order to fully consider the number of pylon points and the point cloud density, the two parameters are standardized. The sum De of the two parameters obtained after processing is used as the distribution characteristic value of the pylon point clouds. Then, a window with height h_2 slides up from the bottom of the pylon at a fixed interval h_1 . If the De value of the layer in the middle of the window is greater than the De value of other layers in the window during the sliding process, the layer is regarded as the one that simultaneously satisfies both the local maximum number of point clouds and the local maximum point cloud density, as shown in Figure 5 (yellow lines).

In practical analysis, not all layers with the above two characteristics are key layers required for subsequent calculations. The top N_1 maximum values are selected to further filter the layers here. Subsequently, important layers can be further determined by calculating the filling rate of each layer. The specific calculation process of the filling rate is as follows: important layers are divided into N_2 grids at a fixed spacing L_1 along the Y' -axis, and the proportion of grids containing points n to the overall count of N_2 grids is defined as the filling rate f , as shown in Figure 6.

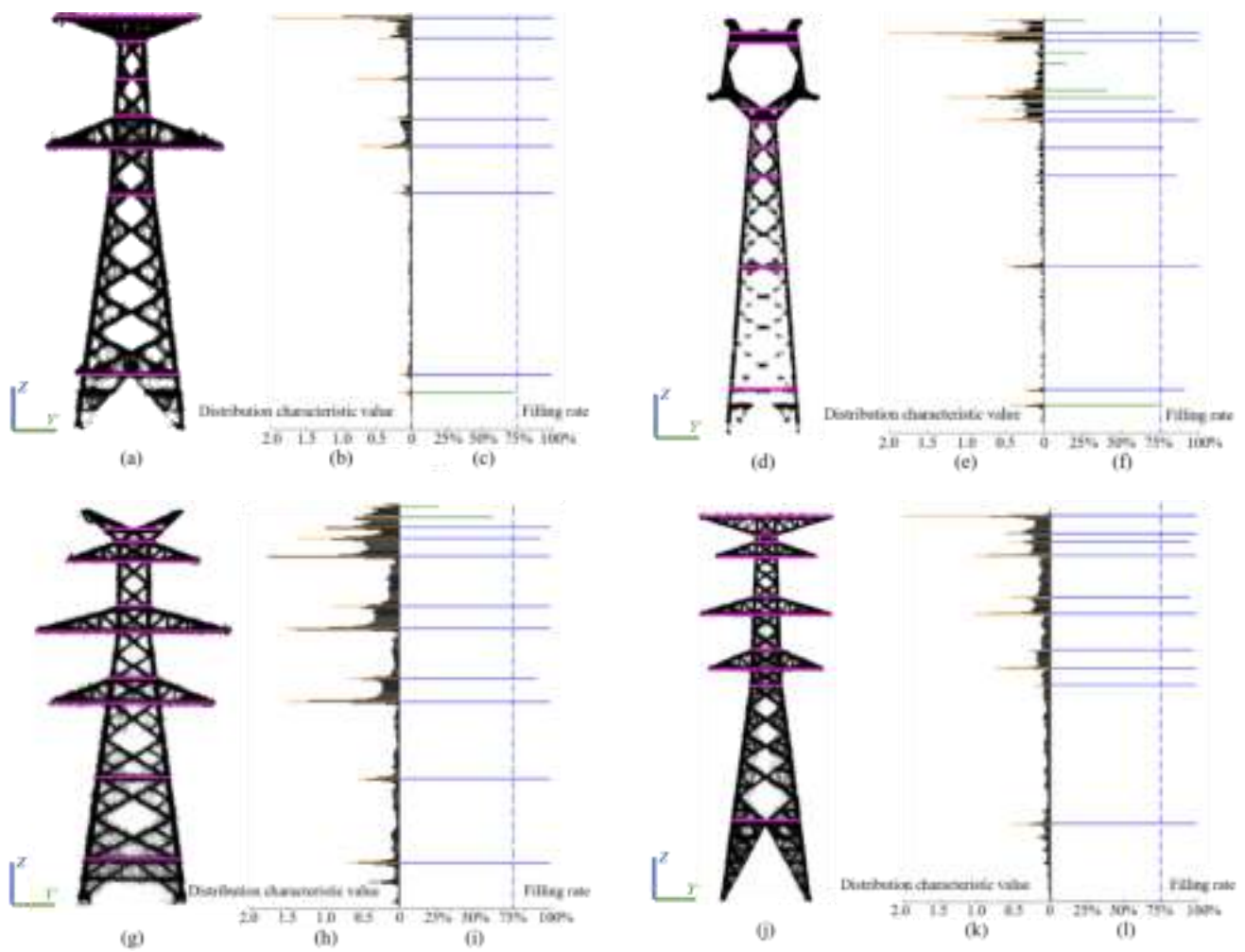


Figure 5. Distribution characteristics of pylon point clouds. (a,d,g,j) The projections of the pylon on the $Y'Z'$ plane. (b,e,h,k) Distribution characteristic value histograms, and yellow lines are layers with both the local maximum number of point clouds and the local maximum density. (c,f,i,l) Filling rate histograms, and blue lines are the key segmentation positions.



Figure 6. The filling rate calculation process.

3.2. Key Segmentation Position Identification

The filling rate f of the expected key segmentation position (yellow lines) is calculated in the previous section. When the filling rate of a layer exceeds the predefined threshold T_f , the key segmentation position S can be described as the average Z coordinate of all the points within the layer, as shown in Figure 5 (blue lines). For type-a pylons and

type-d pylons, all the key segmentation positions can be directly identified using the aforementioned method. However, for the other two types of pylons, besides the positions obtained as described above, additional approaches are required to determine other key segmentation positions.

For the type-b pylon, we first consider the outer contour. Due to the presence of a hollow section in the layer containing the connecting insulator position for the type-b pylons, the filling rate of this part is relatively low and thus not considered as a key segmentation position. Considering that the projected shape of the structure above the pylon head of this type of pylon varies with height in the $X'Y'$ plane, the shape parameter G_i is introduced here to better identify key segmentation positions. This parameter defines the ratio of the maximum projection length of the point clouds on the X' and Y' axes to the minimum value. The sum of G_1 and the error constant C_e serves as the threshold T_G for the shape parameter. Starting from G_1 , each G_i is sequentially compared to the threshold T_G . If G_i exceeds T_G , the point cloud layer corresponding to G_{i-1} is the segmentation position S_{i-1} between the pylon head and pylon body, as shown in Figure 7 (red line). Through observation, it can be found that the two segmentation positions with lower filling rates above the segmentation position are the key segmentation positions supporting the connecting lines. Additionally, the two segmentation positions below the segmentation position with filling rates that meet the requirements need not be considered, and this processing will not affect the subsequent parameter calculation.

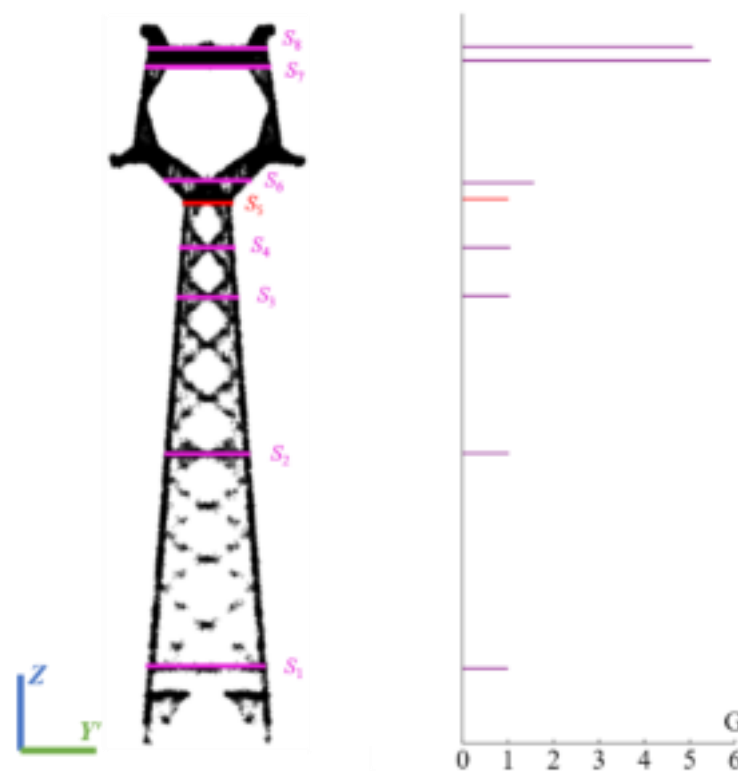


Figure 7. Shape parameters of the type-b pylon.

Before modeling the type-b pylons, it is essential to obtain the dimensions of their internal hollow structures. From Figure 6, it is evident that the hollow position lies between the key segmentation position S_5 of the pylon body and the key segmentation position S_7 of the pylon head. Furthermore, there are relatively lower filling rates at the key segmentation positions inside the hollow structure compared to the aforementioned positions. Combining the positional relationship and point cloud distribution characteristics, the key segmentation positions in the hollow structure can be identified.

For the type-b pylon and the type-c pylon, it can be found that the segmentation position at the top of the pylon has not been identified due to the low filling rate of the point clouds by observing Figure 5, and it needs to be considered separately. The highest among the expected key segmentation positions that do not meet the filling rate threshold requirement can be chosen as one of the key segmentation positions. Finally, all the key segmentation positions of the type-b pylon and the type-c pylon can be obtained, as shown in Figure 8.

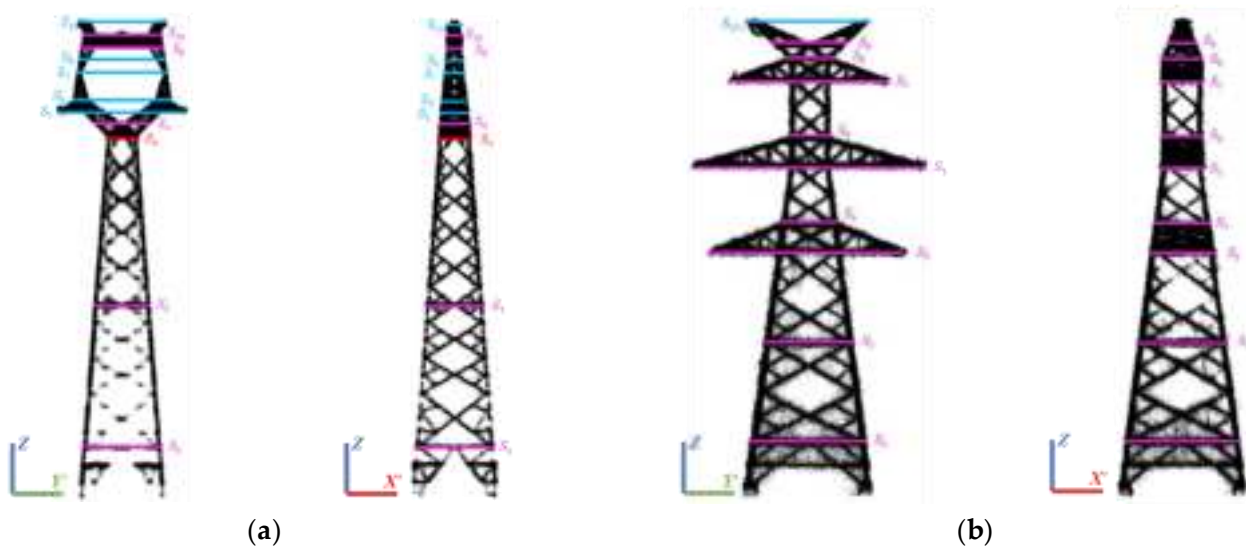


Figure 8. The key segmentation positions of the pylon. (a) Type-b pylon. (b) Type-c pylon.

3.3. Determination of Feature Points

The main purpose of determining pylon feature points is to better calculate various parameters of unknown pylons, such as pylon height, cross-arm length, pylon leg spacing, etc. This study chooses points on the boundaries of pylon key segmentation positions as feature points. The four types of pylons considered in this study have the same pylon body and pylon leg structures. These feature points suffice to calculate the required parameters for these two structures. However, each type of pylon has a complex and unique pylon head structure, requiring the identification of additional feature points to calculate the relevant parameters of the pylon head.

Taking the type-b pylon as an example to introduce the feature point selection strategy, the selected position of the feature points of the type-b pylon is shown in Figure 9, in which the orange points represent the feature points obtained in conjunction with the key segmentation positions. Point 25, as indicated in the figure, represents the connection point between the cross-arm and the pylon body. The Z-axis coordinate can be determined by identifying its characteristic position. Combined with the surrounding point 27 and point 30, a line can be fitted to calculate the Y'-axis coordinate. Similarly, coordinate information on point 11, point 12, and point 26 can be obtained in this way. Regarding point 7 and point 8, the point clouds between point 1 and point 2 are layered along the Y'-axis with a certain interval, and the point with the maximum Z-axis coordinate in each layer is selected. Then, moving from point 1 and point 2 towards the middle by a sliding window, the first points with Z-axis coordinates located at the key segmentation position S_{10} are found. These two points are identified as point 7 and point 8, as indicated by the red markers in Figure 10.

In total, 36 feature points need to be determined for the type-a pylon, while the type-b, type-c, and type-d pylons require 63, 51, and 52 feature points, respectively.

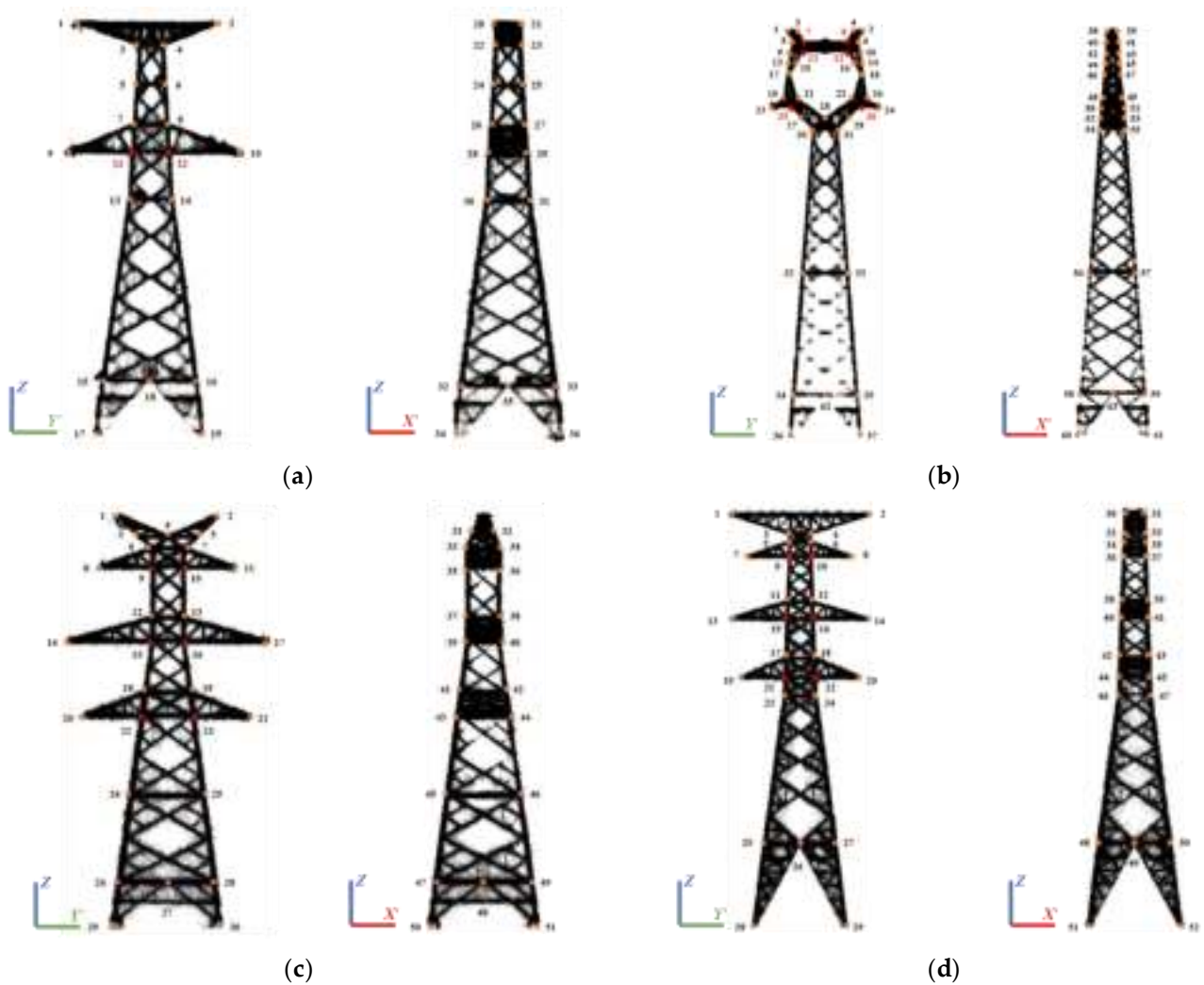


Figure 9. Pylon feature points and numbers, and red points are the feature points obtained by combining with the surrounding orange points. (a) Type-a pylon. (b) Type-b pylon. (c) Type-c pylon. (d) Type-d pylon.

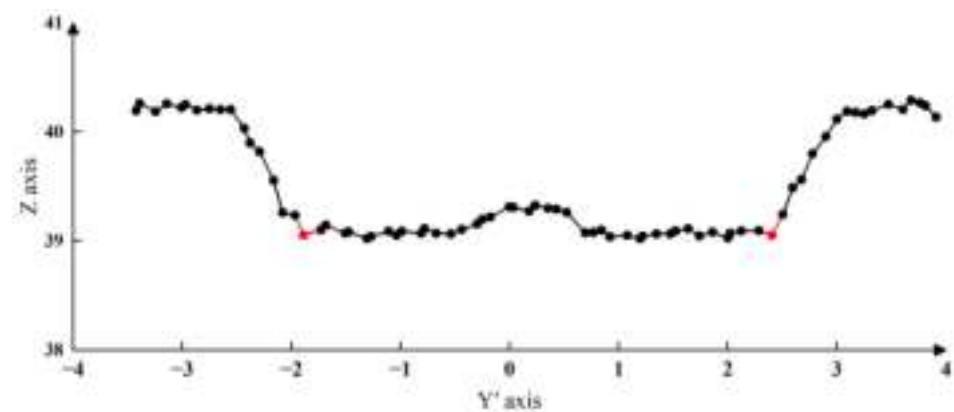


Figure 10. Point cloud distribution curve of the type-b pylon head.

3.4. Calculation of Characteristic Parameters

Based on the coordinate information on the feature points, characteristic parameters are calculated from the perspectives of height, length, and width. This study takes the type-a pylon as an example to introduce the parameters that need to be calculated. In terms of height, the vertical coordinate difference between point 20 and point 30 represents

the pylon head height. The vertical coordinate difference between point 30 and point 32 represents the pylon body height, and the difference between point 32 and point 34 represents the pylon leg height. Concerning length, the ground line cross-arm length can be obtained by measuring the horizontal coordinate difference between point 1 and point 2, while the cross-arm length is determined by the difference between point 9 and point 10. Additionally, the difference in horizontal coordinates between point 17 and point 19 represents the pylon leg spacing. Regarding width, for points projected onto the X'Z plane, calculating the horizontal coordinate difference of the points at the same height can provide the required width parameters.

4. Experiments

In this section, the data and parameters used in the experiment are first introduced in Section 4.1, and then the accuracy of the tower type identification and feature parameter calculation is, respectively, listed in Sections 4.2 and 4.3. Finally, the reconstructed 3D models of the pylons are shown in Section 4.4.

4.1. Experimental Data and Parameter Introduction

This study obtained point cloud data using the DJI M300RTK flight platform with the integrated Livox L1 LiDAR module. The basic parameters of the LiDAR data are shown in Table 1. The collected LiDAR data comprise not only the 3D coordinates of the points but also their RGB color information. The data cover power transmission corridors with voltage levels of 110 kV and 220 kV in Hubei Province and Sichuan Province. In order to facilitate the progress of this study, the data were segmented using the open-source software CloudCompare 2.13 to obtain the pylon point cloud data.

Table 1. Basic parameters of LiDAR data.

Point Density	Horizontal Accuracy	Vertical Accuracy
>100 pts/m ²	10 cm	5 cm

The programs for pylon type identification and parameter calculation were written in Python and run on a laptop. The laptop's configuration information is shown in Table 2. The pylon point cloud data obtained by segmentation were processed using the method proposed in this paper. The parameters involved in the processing are shown in Table 3.

Table 2. Laptop configuration information.

Laptop	CPU	GPU	RAM
Lenovo Legion R9000P 2023	AMD Ryzen 9 7945HX	NVIDIA RTX 4060	16 GB

Table 3. Parameter settings.

Parameters	Meaning	Values
H	Minimum height of point clouds for redirection	$(3/4) \times$ the pylon height
h_1	The layer interval along the Z-axis direction	0.1 m
h_2	The height of the sliding window	1.1 m
L_1	The grid interval along the Y'-axis direction	0.1 m
T_f	The threshold of filling rate	75%
C_e	Error constant	0.5
N_1 (type-a)	The selected number of key layers	8
N_1 (type-b)		14
N_1 (type-c)		11
N_1 (type-d)		10

4.2. Accuracy of Pylon Type Identification

Four different similarity evaluation measurements, namely DTW distance, FastDTW distance, discrete Fréchet distance, and Hausdorff distance, are employed for pylon type identification using the method described in the previous section. This section uses two types of shape curve datasets to conduct the experiments. One is the dataset obtained directly based on drawing information, and the other is the dataset obtained by linearly scaling the above dataset to the same height. Finally, the pylon type identification results of the four similarity evaluation measurements were obtained.

To comprehensively evaluate the identification performance of the various similarity measurements, this study used precision, recall, and F-score as three indexes to analyze the experimental results [25].

Precision refers to the proportion of pylons predicted to belong to a certain type that actually belongs to that type in the experiment.

$$P = \frac{TP}{TP + FP} \times 100\% \quad (9)$$

Recall refers to the proportion of pylons of a certain type that are ultimately predicted to belong to that type.

$$R = \frac{TP}{TP + FN} \times 100\% \quad (10)$$

where TP is the number of samples determined to be of a certain pylon type and actually belonging to that type; FP is the number of samples determined to be of a certain pylon type but actually belonging to other types; and FN is the number of samples of a certain pylon type that are determined to be of other types.

The F-score refers to the harmonic mean of precision and recall, and it was used as an overall evaluation index of pylon identification performance in this study.

$$F = \frac{2P \times R}{P + R} \times 100\% \quad (11)$$

The larger the values of these three evaluation indexes, the better the identification performance of the similarity measurement. For the four types of pylons in this study, the pylon type identification results using different similarity evaluation methods are shown in Figure 11.

Using the original dataset for calculation, regarding precision, the DTW distance and FastDTW distance exhibited the highest precision for both the type-a pylon and the type-b pylon. For the type-c pylon, the precision of the discrete Fréchet distance was 89.7%, which was significantly higher than the other three similarity measurements. For the type-d pylon, all four similarity measurements were basically equivalent. In terms of recall, for the type-a pylon and the type-b pylon, the Hausdorff distance and Discrete Fréchet distance yielded slightly higher results. For the type-c pylon, the recall of all four similarity measurements was around 76%. For the type-d pylon, the results of all four similarity measurements were relatively lower compared to the other three types. Considering the overall evaluation index of the algorithm performance, for the type-a pylon, all four similarity measurements yielded similar results, with the Hausdorff distance slightly inferior. For the type-b pylon, the F-scores of all four similarity measurements were below 66%, indicating poor identification performance. For the type-c pylon, the F-score of the discrete Fréchet distance was 83.8%, which was better than other similarity measurements. For the type-d pylon, the DTW distance and FastDTW distance yielded slightly higher F-scores.

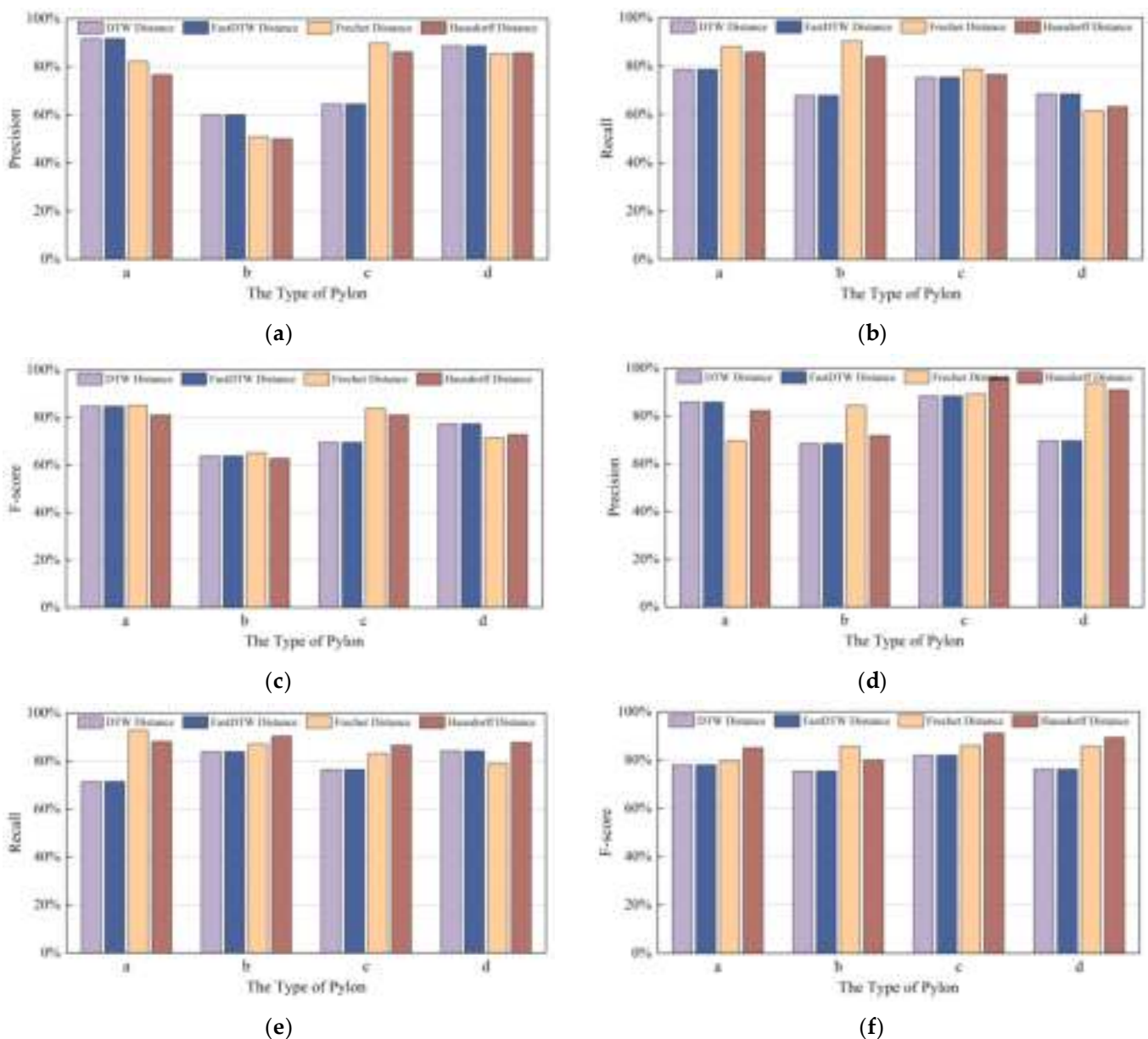


Figure 11. Identification results of pylon types using different similarity measurements. (a) The precision of four similarity measurements (original dataset). (b) The recall of four similarity measurements (original dataset). (c) The F-score of four similarity measurements (original dataset). (d) The precision of four similarity measurements (linearly transformed dataset). (e) The recall of four similarity measurements (linearly transformed dataset). (f) The F-score of four similarity measurements (linearly transformed dataset).

Using the dataset transformed by linear scaling for calculation, all four similarity measurements showed a significant improvement in the precision of identifying the type-b pylon. For the type-c pylon and the type-d pylon, the precision of the Hausdorff distance could exceed 90%. Regarding recall, for the type-a pylon, the discrete Fréchet distance yielded higher results. For the other three types of pylons, the recall results of the Hausdorff distance were 90.3%, 86.5%, and 87.7%, respectively, significantly higher than the other similarity measurements. Considering the overall evaluation index of the algorithm performance, the F-scores of all four similarity measurements had seen substantial improvement for the type-b pylon. For the other three types of pylons, the F-score of the Hausdorff distance was higher than the other three measurements.

In summary, when identifying the type of pylon point clouds, using the ordinary dataset for calculation, the discrete Fréchet distance exhibited the best overall evaluation index, with an average F-score of 76.4%. Using the dataset transformed by linear scaling for calculation, the overall evaluation index of the Hausdorff distance was the best, with an average F-score of 86.4%.

4.3. Calculation Accuracy of Pylon Characteristic Parameters

This section validated the accuracy of calculating key parameters for four types of pylons using practical examples. The type-a pylon was taken as a typical case here. The calculated values of this pylon are listed in Table 4 and were compared with the manual measurement values. Finally, the relative error was obtained.

Table 4. Parameter calculation results of the type-a pylon.

Serial Number	Key Points Connection	Position Description	Calculated Value/m	Manual Measurement Value/m	Relative Error
1	1–2 abscissa difference	Ground line cross-arm length	10.2897	10.5	−2.00%
2	3–4 abscissa difference	/	1.6932	1.8	−5.93%
3	5–6 abscissa difference	/	2.0760	2.1	−1.14%
4	7–8 abscissa difference	/	2.4361	2.4	1.50%
5	9–10 abscissa difference	Cross-arm length	12.5012	13	−3.84%
6	11–12 abscissa difference	/	2.6784	2.7	−0.80%
7	13–14 abscissa difference	/	3.1005	3.2	−3.11%
8	15–16 abscissa difference	/	6.8951	7	−1.50%
9	17–19 abscissa difference	Pylon leg spacing	8.0340	8	0.43%
10	20–21 abscissa difference	Ground line cross-arm width	1.5211	1.6	−4.93%
11	22–23 abscissa difference	/	1.6932	1.8	−5.93%
12	24–25 abscissa difference	/	2.1018	2.1	0.09%
13	26–27 abscissa difference	/	2.4824	2.4	3.43%
14	28–29 abscissa difference	Cross-arm width	2.7438	2.8	−2.01%
15	30–31 abscissa difference	/	3.1937	3.2	−0.20%
16	32–33 abscissa difference	/	6.9042	7	−1.37%
17	34–36 abscissa difference	Pylon leg spacing	8.0179	8	0.22%
18	32–34 ordinate difference	Pylon leg height	4.0512	4	1.28%
19	30–32 ordinate difference	Pylon body height	13.4974	13.5	−0.02%
20	28–30 ordinate difference	/	3.4989	3.5	−0.03%
21	26–28 ordinate difference	/	2.0084	2	0.42%
22	24–26 ordinate difference	/	2.9849	3	−0.50%
23	22–24 ordinate difference	/	3.0140	3	0.47%
24	20–22 ordinate difference	/	1.5022	1.5	0.15%

Finally, it was found that the overall relative error did not exceed 5% by calculating 80 pylon samples and comparing their maximum relative errors, thus validating the feasibility of the algorithm. The calculated results for the pylons are presented in Table 5.

Table 5. Calculation error.

Pylon Type	Quantity	Maximum Relative Error	Average Error
a	20	3.28%	2.36%
b	20	4.96%	2.71%
c	20	4.62%	3.05%
d	20	4.37%	2.93%

4.4. Pylon Reconstruction

Based on the characteristic parameters calculated in the previous section, 3D models of the four types of pylons were reconstructed. The final models are shown in Figure 12.

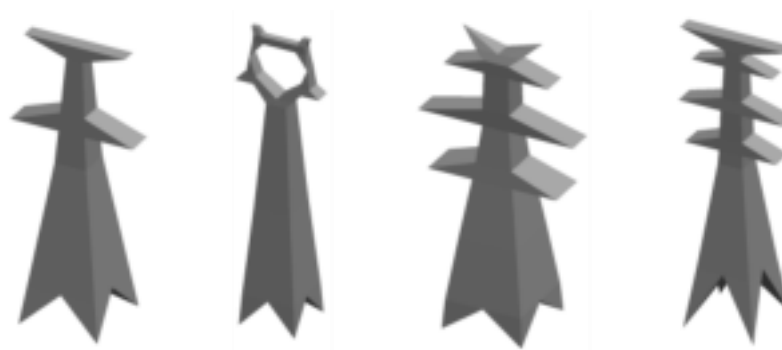


Figure 12. 3D models of pylons.

5. Discussion

This section mainly discusses the errors encountered in pylon type identification, the impact of the LiDAR data noise points on the calculation of characteristic parameters, and the influence of data sparsity.

5.1. Errors in Pylon Type Identification

When using the ordinary dataset, the DTW distance and FastDTW distance tend to misclassify the type-b pylon and the type-d pylon as the type-c pylon. This may be attributed to the fact that DTW is a local matching method insensitive to global shape changes, while pylon shapes involve global changes, resulting in the misidentification of certain pylon types. The discrete Fréchet distance and Hausdorff distance exhibit more errors in identifying the type-d pylon point clouds, which may be related to the similarity in height between the type-d pylon and other types of pylons in the dataset. After using the linearly transformed dataset, the errors in pylon type identification are reduced. Currently, this method is only applicable to identifying existing pylon types in the dataset. Future research can focus on designing more suitable identification algorithms based on the findings of this study to improve its generality.

5.2. The Influence of Noise Points on the Calculation of Characteristic Parameters

Noise points are primarily distributed in two areas of the pylon. One of the areas is at the junction of the pylon body and the cross-arm, as indicated by the blue circle in Figure 13. These noise points cause the selected feature points on the pylon body to shift outward, resulting in excessive errors in characteristic parameter calculation. In this study, the true feature points can be distinguished by the density characteristics of points after the key segmentation positions layering, thereby eliminating the interference of such noise points on feature point selection. Another part of the noise points primarily consists of insulator string points and line points, as shown by the red circle in Figure 13. This kind of noise point has the greatest impact on the calculation of characteristic parameters. If such points exist, it will be difficult to accurately calculate the pylon parameters using the method proposed in this paper. Therefore, it is necessary to manually remove such noise points before identifying the pylon type.

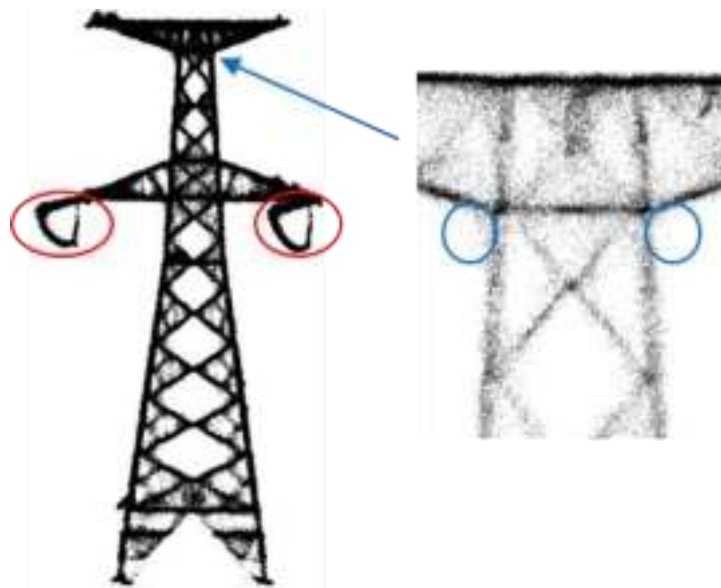


Figure 13. Pylon point clouds with interference points.

5.3. The Influence of Data Sparsity

In the process of power inspection, the change in flight height and speed of the UAV will lead to the difference in point density. When the flight altitude is higher and the flight speed is faster, the obtained point density is lower. To analyze the influence of data sparsity on pylon type identification and characteristic parameter calculation, the original pylon point cloud data are sampled by the voxel sampling method. The number of pylon points obtained at different sampling distances is shown in Table 6. When the sampling distance is less than 0.4 m, the methods in this paper can be used to correctly judge the type of pylon and the calculated pylon parameters are relatively accurate. However, when the sampling distance is greater than 0.4 m, the point clouds become sparse and the distribution parameters of the point clouds cannot reflect the characteristics of the key segmentation positions of the pylons. The redirection processing is also affected when the sampling distance exceeds 0.6 m. The parameters in Table 3 cannot make the horizontal direction of the pylon perpendicular to the X-axis. In this case, the parameters need to be reset to solve this problem.

Table 6. The number of pylon points sampled with different distances.

Pylon Type	The Number of Points				
	Original Point Cloud	Sample Distance			
		0.1 m	0.2 m	0.3 m	0.4 m
a	203,792	67,631	20,809	9528	5492
b	102,429	47,194	16,360	7786	4706
c	239,345	128,429	42,751	19,832	11,063
d	196,601	112,847	42,108	19,851	11,166

6. Conclusions

This study proposes an improved method for identifying power pylon types and a novel method for the automatic calculation of characteristic parameters, aiming to solve the problems of complex calculation and low efficiency in existing methods. They can provide the necessary data support for reconstructing 3D models of pylons. The proposed method in this paper exhibits several characteristics and demonstrates great potential in utilizing

airborne LiDAR data to acquire basic information about pylons. The research results of this article can be summarized as follows.

(1) This article introduces a method for generating pylon shape curves based on point cloud data. On this basis, an improved method for point cloud type identification based on similarity measurements and a linearly transformed dataset is proposed. This method simplifies the pylon type identification problem in 3D space to curve identification in 2D space. It can effectively identify a variety of pylon types and provide information support for the parameter calculation of pylon point clouds.

(2) This study compared the identification effects of four similarity measurements: the DTW distance, FastDTW distance, discrete Fréchet distance, and Hausdorff distance. In terms of the overall evaluation index (F-score), when using the ordinary dataset, the discrete Fréchet distance as the similarity measurement yielded the optimal overall evaluation index of 76.4%. Meanwhile, the Hausdorff distance as the similarity measurement achieved the best performance using the dataset after linear transformation, with an average F-score of 86.4%.

(3) A novel method for calculating pylon parameters based on point cloud distribution characteristics is proposed. This method can effectively extract point cloud specific information and ensure the accuracy of the parameter calculation. Through the calculation and analysis of 80 groups of pylons, it could be found that the maximum relative error produced by this algorithm did not exceed 5%, thus verifying the feasibility of the algorithm.

Although the method proposed in this study can yield relatively accurate results in pylon type identification and characteristic parameter calculation, there are still some aspects that need to be optimized in future research, as follows. (1) The pylon types considered in this paper are limited by the dataset. Therefore, it is necessary to expand the dataset in future studies. (2) The selection method for feature points needs to be continuously optimized to reduce calculation errors in the characteristic parameters. (3) In future research, pylons with asymmetric structure will also be taken into account.

Author Contributions: Conceptualization, S.Z. and L.W.; methodology, S.Z.; validation, S.Z.; resources, L.W. and B.S.; data curation, S.Z., S.W. and G.W.; writing—original draft preparation, S.Z.; writing—review and editing, S.Z.; visualization, S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lu, Z.; Gong, H.; Jin, Q.; Hu, Q.; Wang, S. A Transmission Tower Tilt State Assessment Approach Based on Dense Point Cloud from UAV-Based LiDAR. *Remote Sens.* **2022**, *14*, 408. [\[CrossRef\]](#)
2. Li, W.; Luo, Z.; Xiao, Z.; Chen, Y.; Wang, C.; Li, J. A GCN-Based Method for Extracting Power Lines and Pylons from Airborne LiDAR Data. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [\[CrossRef\]](#)
3. Chen, C.; Jin, A.; Yang, B.; Ma, R.; Sun, S.; Wang, Z.; Zong, Z.; Zhang, F. DCPLD-Net: A Diffusion Coupled Convolution Neural Network for Real-Time Power Transmission Lines Detection from UAV-Borne LiDAR Data. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102960. [\[CrossRef\]](#)
4. Yang, J.; Kang, Z. Voxel-Based Extraction of Transmission Lines from Airborne LiDAR Point Cloud Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 3892–3904. [\[CrossRef\]](#)
5. Xie, X.; Liu, Z.; Xu, C.; Zhang, Y. A Multiple Sensors Platform Method for Power Line Inspection Based on a Large Unmanned Helicopter. *Sensors* **2017**, *17*, 1222. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Jiang, S.; Jiang, W.; Huang, W.; Yang, L. UAV-Based Oblique Photogrammetry for Outdoor Data Acquisition and Offsite Visual Inspection of Transmission Line. *Remote Sens.* **2017**, *9*, 278. [\[CrossRef\]](#)
7. Yang, L.; Fan, J.; Liu, Y.; Li, E.; Peng, J.; Liang, Z. A Review on State-of-the-Art Power Line Inspection Techniques. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 9350–9365. [\[CrossRef\]](#)
8. Zhou, R.; Jiang, W.; Jiang, S. A Novel Method for High-Voltage Bundle Conductor Reconstruction from Airborne LiDAR Data. *Remote Sens.* **2018**, *10*, 2051. [\[CrossRef\]](#)

9. Zhao, P.; Yang, W.; Feng, Y.; Li, F.; Huang, X. Construction of 3D Scene of Transmission Line Corridor Based on GIM and 3D GIS. *J. Phys. Conf. Ser.* **2021**, *2005*, 012083. [\[CrossRef\]](#)
10. Li, Q.; Chen, Z.; Hu, Q. A Model-Driven Approach for 3D Modeling of Pylon from Airborne LiDAR Data. *Remote Sens.* **2015**, *7*, 11501–11524. [\[CrossRef\]](#)
11. Tang, Q.; Zhang, L.; Lan, G.; Shi, X.; Duanmu, X.; Chen, K. A Classification Method of Point Clouds of Transmission Line Corridor Based on Improved Random Forest and Multi-Scale Features. *Sensors* **2023**, *23*, 1320. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Zhu, S.; Li, Q.; Zhao, J.; Zhang, C.; Zhao, G.; Li, L.; Chen, Z.; Chen, Y. A Deep-Learning-Based Method for Extracting an Arbitrary Number of Individual Power Lines from UAV-Mounted Laser Scanning Point Clouds. *Remote Sens.* **2024**, *16*, 393. [\[CrossRef\]](#)
13. Guo, B.; Huang, X.; Li, Q.; Zhang, F.; Zhu, J.; Wang, C. A Stochastic Geometry Method for Pylon Reconstruction from Airborne LiDAR Data. *Remote Sens.* **2016**, *8*, 243. [\[CrossRef\]](#)
14. Yang, L.; Kong, S.; Deng, J.; Li, H.; Liu, Y. DRA-Net: A Dual-Branch Residual Attention Network for Pixelwise Power Line Detection. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 5010813. [\[CrossRef\]](#)
15. Wang, G.; Wang, L.; Wu, S.; Zu, S.; Song, B. Semantic Segmentation of Transmission Corridor 3D Point Clouds Based on CA-PointNet++. *Electronics* **2023**, *12*, 2829. [\[CrossRef\]](#)
16. Shen, Y.; Yang, Y.; Jiang, J.; Wang, J.; Huang, J.; Ferreira, V.; Chen, Y. A Novel Method to Segment Individual Wire from Bundle Conductor Using UAV-LiDAR Point Cloud Data. *Measurement* **2023**, *211*, 112603. [\[CrossRef\]](#)
17. Han, W. Three-dimensional power tower modeling with airborne LiDAR data. *J. Yangtze River Sci. Res. Inst.* **2012**, *29*, 122–126. [\[CrossRef\]](#)
18. Zhou, R.; Jiang, W.; Huang, W.; Xu, B.; Jiang, S. A Heuristic Method for Power Pylon Reconstruction from Airborne LiDAR Data. *Remote Sens.* **2017**, *9*, 1172. [\[CrossRef\]](#)
19. Camuffo, E.; Mari, D.; Milani, S. Recent Advancements in Learning Algorithms for Point Clouds: An Updated Overview. *Sensors* **2022**, *22*, 1357. [\[CrossRef\]](#)
20. Qiao, Y.; Xi, X.; Nie, S.; Wang, P.; Guo, H.; Wang, C. Power Pylon Reconstruction from Airborne LiDAR Data Based on Component Segmentation and Model Matching. *Remote Sens.* **2022**, *14*, 4905. [\[CrossRef\]](#)
21. Chen, S.; Wang, C.; Dai, H.; Zhang, H.; Pan, F.; Xi, X.; Yan, Y.; Wang, P.; Yang, X.; Zhu, X.; et al. Power Pylon Reconstruction Based on Abstract Template Structures Using Airborne LiDAR Data. *Remote Sens.* **2019**, *11*, 1579. [\[CrossRef\]](#)
22. Silva, F.; Amaro, N. Transmission Tower Classification Using Point Cloud Similarity. In Proceedings of the APCA International Conference on Automatic Control and Soft Computing, Caparica, Portugal, 6–8 July 2022; pp. 609–618.
23. Chen, Z.; Lan, Z.; Long, H.; Hu, Q. 3D Modeling of Pylon from Airborne LiDAR Data. In Proceedings of the Remote Sensing of the Environment: 18th National Symposium on Remote Sensing of China, Wuhan, China, 14 May 2014; p. 915807.
24. Wang, H.; Hu, T.; Wang, Z.; Kang, Z.; Akwensi, P.H.; Yang, J. Reconstruction of Power Pylons from LiDAR Point Clouds Based on Structural Segmentation and Parameter Estimation. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [\[CrossRef\]](#)
25. Zhang, M.; Su, X.; Xu, H.; Li, H.; Wang, D. Transmission Tower Category Identification from Airborne LiDAR Point Clouds Based on Shape Curve. In Proceedings of the 3rd International Conference on Mechatronics, Automation and Intelligent Control, Guilin, China, 15–17 September 2023; p. 012026. [\[CrossRef\]](#)
26. Principi, E.; Squartini, S.; Cambria, E.; Piazza, F. Acoustic Template-Matching for Automatic Emergency State Detection: An ELM Based Algorithm. *Neurocomputing* **2015**, *149*, 426–434. [\[CrossRef\]](#)
27. Obaid, M.; Hodrob, R.; Abu Mwais, A.; Aldababsa, M. Small Vocabulary Isolated-Word Automatic Speech Recognition for Single-Word Commands in Arabic Spoken. *Soft Comput.* **2023**, 1–14. [\[CrossRef\]](#)
28. Salvador, S.; Chan, P. Toward Accurate Dynamic Time Warping in Linear Time and Space. *IDA* **2007**, *11*, 561–580. [\[CrossRef\]](#)
29. Gao, Y.; Yang, Y.; Ma, Y.; Xu, W. Study on Intelligent Diagnosis of Railway Turnout Switch Based on Improved FastDTW and Time Series Segmentation under Big Data Monitoring. *Math. Probl. Eng.* **2022**, *2022*, 7048813. [\[CrossRef\]](#)
30. Yeo, K.; Yin, O.S.; Han, P.Y.; Kwee, W.K. Real Time Mobile Application of In-Air Signature with Fast Dynamic Time Warping (FastDTW). In Proceedings of the 2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, Malaysia, 19–21 October 2015; pp. 315–320.
31. Barbay, J. Adaptive Computation of the Discrete Fréchet Distance. In Proceedings of the String Processing and Information Retrieval, Lima, Peru, 14 September 2018; pp. 50–60.
32. Avraham, R.B.; Filtser, O.; Kaplan, H.; Katz, M.J.; Sharir, M. The Discrete and Semicontinuous Fréchet Distance with Shortcuts via Approximate Distance Counting and Selection. *ACM Trans. Algorithms* **2015**, *11*, 1–29. [\[CrossRef\]](#)
33. Ali, M.; Hussain, Z.; Yang, M.-S. Hausdorff Distance and Similarity Measures for Single-Valued Neutrosophic Sets with Application in Multi-Criteria Decision Making. *Electronics* **2022**, *12*, 201. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Multi-Object Vehicle Detection and Tracking Algorithm Based on Improved YOLOv8 and ByteTrack

Longxiang You, Yajun Chen *, Ci Xiao, Chaoyue Sun and Rongzhen Li

School of Electronic Information Engineering, China West Normal University, Nanchong 637009, China; ylxzxx@stu.cwnu.edu.cn (L.Y.); xiaoci035560@stu.cwnu.edu.cn (C.X.); sunchaoyue@stu.cwnu.edu.cn (C.S.); lee1225rz@stu.cwnu.edu.cn (R.L.)

* Correspondence: scncjy@cwnu.edu.cn

Abstract: Vehicle detection and tracking technology plays a crucial role in Intelligent Transportation Systems. However, due to factors such as complex scenarios, diverse scales, and occlusions, issues like false detections, missed detections, and identity switches frequently occur. To address these problems, this paper proposes a multi-object vehicle detection and tracking algorithm based on CDS-YOLOv8 and improved ByteTrack. For vehicle detection, the Context-Guided (CG) module is introduced during the downsampling process to enhance feature extraction capabilities in complex scenarios. The Dilated Reparam Block (DRB) is reconstructed to tackle multi-scale issues, and Soft-NMS replaces the traditional NMS to improve performance in densely populated vehicle scenarios. For vehicle tracking, the state vector and covariance matrix of the Kalman filter are improved to better handle the nonlinear movement of vehicles, and Gaussian Smoothed Interpolation (GSI) is introduced to fill in trajectory gaps caused by detection misses. Experiments conducted on the UA-DETRAC dataset show that the improved algorithm increases detection performance, with mAP@0.5 and mAP@0.5:0.95 improving by 9% and 8.8%, respectively. In terms of tracking performance, mMOTA improves by 6.7%. Additionally, comparative experiments with mainstream detection and two-stage tracking algorithms demonstrate the superior performance of the proposed algorithm.



Citation: You, L.; Chen, Y.; Xiao, C.; Sun, C.; Li, R. Multi-Object Vehicle Detection and Tracking Algorithm Based on Improved YOLOv8 and ByteTrack. *Electronics* **2024**, *13*, 3033. <https://doi.org/10.3390/electronics13153033>

Academic Editor: Fernando De la Prieta Pintado

Received: 11 July 2024

Revised: 29 July 2024

Accepted: 30 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: computer vision; intelligent transportation systems; vehicle detection; vehicle tracking

1. Introduction

Vehicle detection and tracking technology plays a key role in Intelligent Transportation Systems [1]. Utilizing this technology enables the real-time monitoring of road traffic flow, assisting traffic management authorities in making informed decisions to effectively alleviate traffic congestion. Additionally, vehicle detection and tracking technology can identify traffic violations, reducing the likelihood of traffic accidents. This technology also provides intelligent perception of vehicles and road environments, offering smarter and more efficient solutions for traffic management. In recent years, AI technology has played a significant role in enhancing the capabilities of vehicular networks. For example, in vehicular edge network computing, Y Ju et al. [2] have conducted research on secure offloading using non-orthogonal multiple access (NOMA) technology and asynchronous deep reinforcement learning. This not only improves the security and efficiency of data transmission but also supports the real-time requirements for vehicle detection and tracking.

However, several challenges are often encountered when performing vehicle detection and tracking. Factors such as complex lighting variations, weather impacts, the diversity of vehicle types and sizes, and high-density traffic scenes can significantly reduce the effectiveness of detection and tracking. In response to these challenges, numerous research efforts have been undertaken. For instance, D. Roy et al. [3] improved vehicle detection and tracking performance by introducing multi-modal information. M. Humayun et al. [4], based on YOLOv4, addressed variations in various weather conditions by adding a Spatial Pyramid Pooling Block. P. Deshmukh et al. [5] used the Swin Transformer to extract

multi-scale features and utilized a Bidirectional Feature Pyramid Network to handle the diversity in vehicle scales. J. Wang et al. [6], based on YOLOv5, introduced an attention mechanism to enhance the detection performance of small vehicles and employed the JDE algorithm for vehicle tracking. T. Bui et al. [7], also based on YOLOv5, introduced an Attention-based Intra-scale Feature Interaction module to accelerate the convergence of the detection network and improved ResNet36 as the DeepSORT feature extraction network for vehicle tracking. In addition to improving object detection models and vehicle tracking methods based on the Kalman filter, several technological solutions have been proposed to address these challenges. Y. Wu et al. [8] proposed a technique using a generative adversarial network (GAN) to convert daytime vehicle datasets into nighttime datasets, enhancing nighttime detection performance from the perspective of dataset expansion. B. Xu et al. [9] introduced a monocular vision framework that uses an image inpainting model to restore occluded vehicles through masking, improving performance in congested traffic scenarios. Additionally, B. Xu et al. [10] employed a method combining key point detection and multiple deep learning models, optimizing the detection and tracking of vehicle axle load points by integrating the YOLO-v5 and HourglassNet models with a dynamics model. H. Xu et al. [11] proposed a Cross-Domain Car Detection Model with an integrated convolutional block attention mechanism (CDCDMA) vehicle detection model that builds a cross-domain object detection framework, emphasizing headlight features to handle changes in complex scenes. C. Zhang et al. [12] introduced a system framework that addresses cross-view vehicle re-identification challenges by learning high-order relationships and topological information. This framework effectively improves recognition accuracy across different views through the feature extraction module, graph convolution module, and graph matching module.

In the realm of vehicle detection, most methods are typically optimized for single scenarios. To address this, this paper proposes an improved algorithm based on YOLOv8 [13]—CDS-YOLOv8. This algorithm introduces the Context-Guided module [14] during the downsampling process to enhance feature extraction capabilities, thereby adapting to complex scene variations. Simultaneously, the Dilated Reparam Block [15] structure is reconstructed and integrated with the C2f module to fully extract multi-scale and sparse features, enhancing the model's performance in multi-scale object detection. Additionally, in the post-processing stage, the algorithm adopts Soft-NMS to replace traditional NMS, preserving more valid candidate boxes in densely populated vehicle scenes through smooth confidence decay. In terms of vehicle tracking, most tracking algorithms typically focus only on high-confidence detection boxes while ignoring low-confidence ones, leading to missed detections of some true targets. To address this, this paper selects ByteTrack [16] as the base tracker and makes adjustments to the state vector and covariance matrix in the Kalman filter. Furthermore, Gaussian Smoothed Interpolation [17] is introduced to post-process vehicle trajectories, compensating for missing trajectories.

The main contributions of this paper are summarized as follows:

1. We proposed an improved vehicle detection model, CDS-YOLOv8: To address the limitations of YOLOv8 in urban environments for vehicle detection, the CDS-YOLOv8 model introduces a CG module during downsampling to enhance feature extraction capabilities in response to complex scene changes. It reconstructs the DRB to better handle multi-scale issues caused by varying camera distances and vehicle sizes. Additionally, Soft-NMS replaces traditional NMS to retain more valid candidate boxes, improving performance in densely populated vehicle scenes.
2. We improved the ByteTrack tracker: ByteTrack was selected as the base tracker. To address the limitations of the original Kalman filter in capturing vehicle motion, improvements were made to the state vector and covariance matrix to better handle the nonlinear changes in vehicle movement. To mitigate track interruptions due to detection loss, GSI was used to fill in tracking gaps, improving track continuity and reducing identity switches and track loss.

3. The integration of the improved detector and tracker was achieved: CDS-YOLOv8 and the improved ByteTrack were combined and validated on public datasets. Experimental results indicate that this method demonstrates advanced performance and superiority across multiple metrics.

2. Related Work

2.1. Vehicle Detection

In recent years, with the rapid development of deep learning technology, significant progress has been made in the field of object detection. Deep learning-based object detection methods can be divided into one-stage and two-stage approaches. The advantage of one-stage methods lies in their fast processing speed, as they directly generate object categories and bounding boxes through a single network. In contrast, two-stage methods first generate candidate regions and then perform fine detection, with the advantage of higher detection accuracy. Representative one-stage methods include the YOLO series, SSD [18], and RetinaNet [19], while representative two-stage methods include Faster R-CNN [20] and Cascade R-CNN [21]. The working principle of the YOLO series is to divide the input image into a fixed-size grid, with each grid directly predicting the location and category probability of the object. SSD extracts the bounding boxes and categories of objects on feature maps of different scales, achieving multi-scale detection. RetinaNet employs a Feature Pyramid Network for feature extraction and introduces Focal Loss to address the issue of class imbalance. Faster R-CNN first generates candidate regions through a Region Proposal Network and then performs fine object classification and bounding box regression on these regions. Cascade R-CNN, building on Faster R-CNN, introduces a multi-stage regression and classification structure that progressively refines candidate regions, further improving detection accuracy.

To meet the practical needs of vehicle detection, several research efforts have further optimized general detectors. Z. Chen et al. [22] proposed an improved method based on SSD by introducing channel attention mechanisms and deconvolution modules for feature fusion, thereby enhancing vehicle detection performance. L. Kang et al. [23] presented a Type-1 Fuzzy Attention method based on YOLOv5 to improve vehicle detection. This approach optimizes detection performance by focusing the detector on the target center. M. Bie et al. [24] proposed a Bidirectional Feature Pyramid Network based on YOLOv5 to enhance the model's feature extraction capabilities. B. Wang et al. [25] introduced an improved method based on YOLOv8 by incorporating the BiFormer attention mechanism and BiFPN network to enhance feature fusion capabilities.

In terms of selecting detection algorithms, YOLOv8 achieves a good balance between speed, accuracy, and model size due to its fast detection speed, high accuracy, and compact model size. Therefore, this paper proposes a CDS-YOLOv8 algorithm based on YOLOv8 to address vehicle detection issues under complex weather conditions, scale diversity, and dense vehicle environments. Without increasing computational load, this algorithm effectively improves vehicle detection performance.

2.2. Multi-Target Vehicle Tracking

Detection-based multi-target tracking algorithms can be divided into single-stage and two-stage methods. Single-stage methods generally use an end-to-end structure, merging object detection and association into a single step to directly produce detection and tracking results, with the advantage of strong real-time performance. Two-stage methods first perform object detection and then carry out object association, offering higher accuracy. Typical single-stage methods include JDE [26] and FairMOT [27]. JDE completes object detection and embedding in the same network, with the detection and embedding parts sharing most of the network parameters. FairMOT integrates object detection and ReID networks into one network but processes detection and re-identification tasks through independent branches.

Typical two-stage tracking algorithms include the SORT series. SORT [28], proposed by A. Bewley et al., uses Faster R-CNN as the detector in the detection stage to output object location and category information. The tracking stage completes trajectory association by combining the Kalman filter and the Hungarian algorithm. Subsequently, N. Wojke et al. proposed the DeepSORT [29] algorithm, which, compared to SORT, introduces the ReID method, using a feature extraction network to obtain the appearance features of objects and performing cascaded matching of appearance features and Intersection Over Union (IoU) during data association. StrongSORT further enhances appearance feature extraction, motion noise handling, and camera motion compensation based on DeepSORT, and supplements and optimizes trajectories with missing detections and associations. Y. Zhang et al. [16] proposed the ByteTrack algorithm, which improves tracking performance by associating almost all detection boxes. Following this, N. Aharon et al. proposed the BoT-SORT [30] algorithm, which improves the Kalman filter on the basis of ByteTrack, adding camera motion compensation and the ReID method.

In vehicle tracking scenarios, Z. Zhao et al. [31] designed a new joint scoring strategy based on the JDE framework and introduced CBAM and Transformer encoders to enhance detection capabilities, thereby improving tracking performance. K. Zhang et al. [32] incorporated the CAENet attention mechanism into YOLOX to improve vehicle detection and used DeepSORT for vehicle tracking. G. Han et al. [33] tracked vehicles based on FairMOT, optimized the DLA-34 network to enhance detection performance, and introduced a joint loss function to reduce identity switching issues.

Compared to single-stage tracking algorithms, two-stage tracking algorithms are more flexible and achieve higher accuracy in complex scenarios. Therefore, this paper opts to improve ByteTrack based on CDS-YOLOv8 for vehicle detection and tracking. Unlike most tracking methods that ignore low-score detection boxes, ByteTrack fully considers low-score detection boxes in trajectory association, effectively addressing the issue of missing trajectory fragments. For instance, when partial occlusion occurs, the detection box scores may vary from high to low. By considering low-confidence score boxes, ByteTrack can identify partially occluded targets, reducing the incidence of missed detections.

3. Methods

In the process of vehicle detection and tracking, the CDS-YOLOv8 detector first outputs the detection boxes for the current frame. The improved Kalman filter then predicts the bounding boxes for the current frame through tracklet prediction. Subsequently, the detection boxes and predicted boxes are matched using the Hungarian Algorithm, with IoU serving as the basis for association. The specific process of secondary matching is illustrated in Figure 1.

The detection boxes output using CDS-YOLOv8 are divided into high-confidence boxes and low-confidence boxes based on their confidence scores, followed by two rounds of association processing. In the first round of association, the improved Kalman filter predicts the tracked tracklets and the lost tracklets and then matches the high-confidence boxes with the predicted boxes. If the match is successful, the tracklet state is updated; if not, the tracked tracklets that are not matched with high-confidence boxes undergo a second round of association with low-confidence boxes. In the second round of association, if the match is successful, the tracklet state is updated; if not, the tracklet enters the lost state. In tracklet management, tracklets that remain unmatched for more than 30 frames are deleted. New tracklets are assigned to unmatched high-confidence detection boxes. Finally, after vehicle detection and tracking, GSI is used to fill in gaps in the trajectories caused by missing detections, optimizing the continuity and completeness of the trajectories.

The CDS-YOLOv8 model is composed of three parts: the Backbone, Neck, and Head, as illustrated in Figure 2:

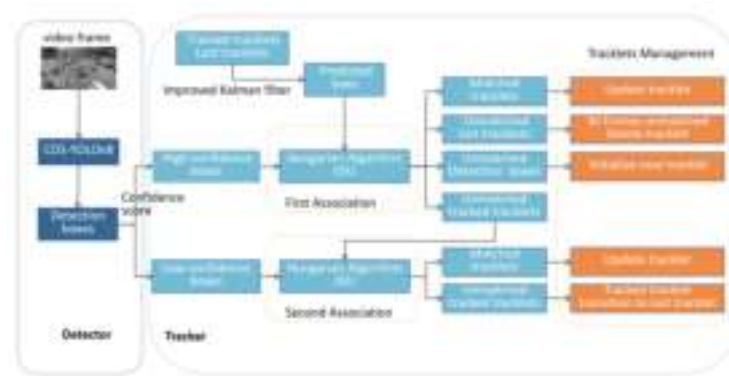


Figure 1. Vehicle Detection and Tracking Flowchart.

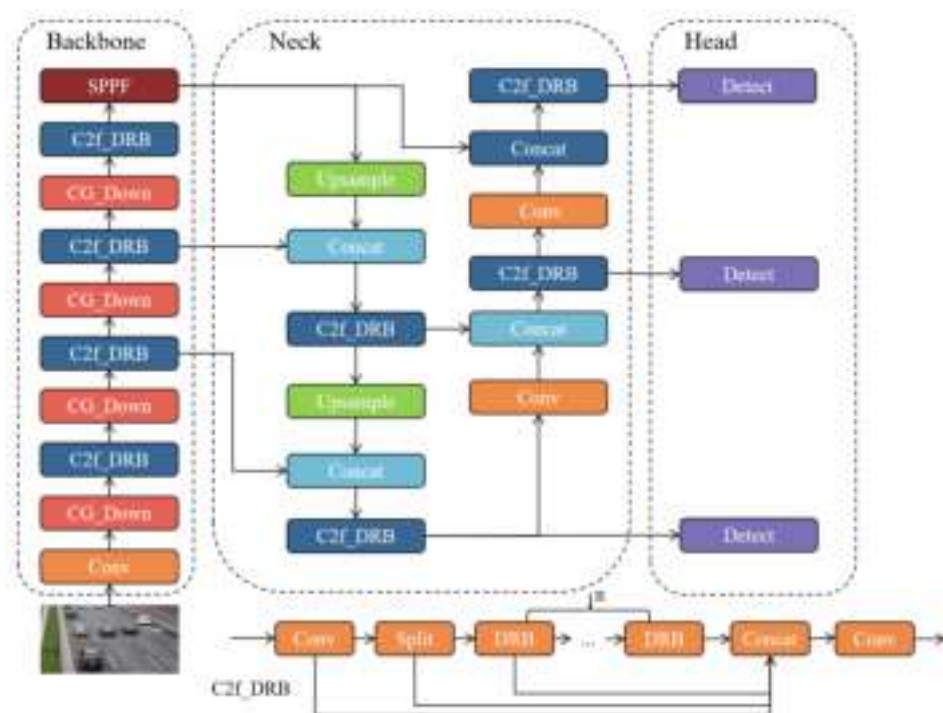


Figure 2. Structure of CDS-YOLOv8.

In Figure 2, the Backbone is used for feature extraction, extracting multi-level features from the input image. The Neck is responsible for feature fusion, combining deep and shallow features using FPN and PAN. The Head contains three detection heads, which handle feature maps of different scales and output the position and category information of the detection boxes. The CDS-YOLOv8 addresses the shortcomings of YOLOv8 in urban vehicle detection with the following improvements: The CG module is introduced during downsampling, which enhances feature extraction in complex environments by combining local and surrounding contextual information and utilizing global information for weighting. In the Backbone and Neck, the DRB structure is restructured and integrated with the C2f module to better address scale diversity issues. In the post-processing stage, Soft-NMS replaces traditional NMS, avoiding the aggressive elimination of candidate boxes and improving detection performance under partial occlusion.

3.1. Context-Guided Downsampling

The environments where vehicles operate include complex scenarios such as nights, cloudy weather, sunny days, and rainy days. To enhance feature extraction capabilities in these complex environments, the Context-Guided module is introduced during the

downsampling process. The CG Module comprises several key components: The Local Feature Extractor uses standard convolution to extract local features of the image, providing fine-grained features and boundary information. The Surrounding Context Extractor employs Dilated Convolution to expand the receptive field of local features, thereby capturing a wider range of contextual information. The Joint Feature Extractor concatenates local features and surrounding contextual features and processes them with Batch Normalization (BN) and SiLU activation, forming joint features that enhance nonlinear expression capabilities. The Global Context Extractor aggregates global context information of the entire image through Global Average Pooling and further processes it using a Multi-Layer Perceptron. This applies weights to the joint features, amplifying useful information while suppressing irrelevant information. The structure of the CG Module is illustrated in Figure 3:

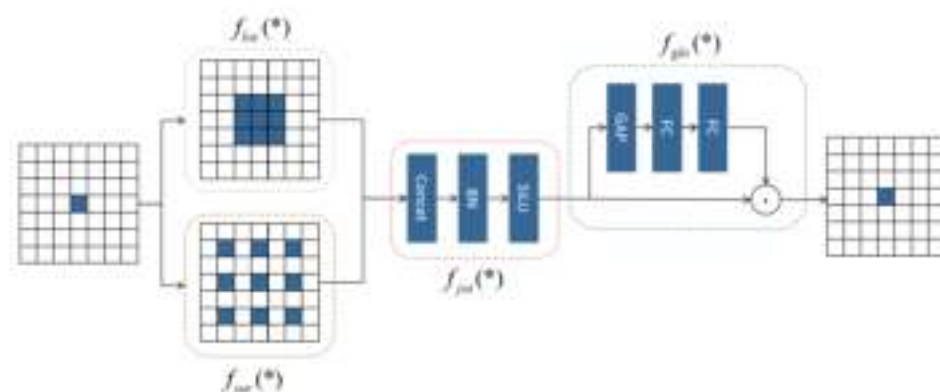


Figure 3. Structure of the Context-Guided module.

In detail, first, the input feature map undergoes 3×3 convolution to extract local features, and 3×3 convolution with a dilation rate of 2 to extract surrounding contextual features. Next, the Joint Feature Extractor fuses the local features and surrounding contextual features along the channel dimension, followed by the BN layer and SiLU activation. Finally, the Global Context Extractor leverages the image's global information, using Global Average Pooling to generate a global feature vector for each channel. This vector is then subjected to nonlinear transformation through two fully connected layers, producing channel weights. These weights are applied to the joint features for weighted optimization, enhancing important features while suppressing unimportant ones, thus boosting the model's feature extraction capability.

3.2. Reconstructed Dilated Reparam Block

In vehicle detection, significant scale variations arise due to differences in camera distances and vehicle sizes. To address this issue, we have reconstructed the Dilated Reparam Block and integrated it into the C2f structure. The DRB structure comprises a large-kernel convolution layer and multiple dilated small-kernel convolution layers. The large-kernel convolution layer captures broader spatial features, while the dilated small-kernel convolution layers capture sparse spatial features through dilation operations. The outputs of these layers are processed through the BN layer and then summed up. Using reparameterization techniques, they are equivalently represented as a single large-kernel convolution layer. This design has two primary advantages: The small-kernel convolution layers, by incorporating dilation, capture features over a larger range. When combined with the large-kernel convolution layer, the receptive field is effectively expanded, eliminating the need to increase the model depth or design larger convolution kernels. During inference, reparameterization techniques merge multiple convolution layers into an equivalent single convolution kernel, maintaining the kernel size and computational complexity while significantly enhancing the model's feature extraction capability. Figure 4 illustrates the DRB structure.

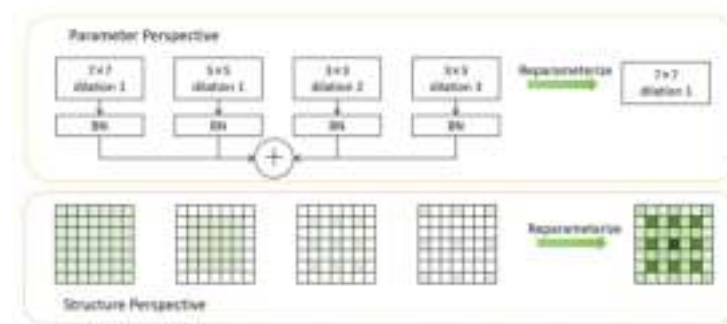


Figure 4. Structure of the Dilated Reparam Block.

From a parameter perspective, the DRB consists of a 7×7 large-kernel convolution layer and three small-kernel convolution layers with different dilation rates. The outputs of all convolution layers are processed through BN layers and are equivalently represented as a single 7×7 large-kernel convolution layer during inference, without significantly increasing the computational load.

Structurally, Figure 4 demonstrates how each convolution layer corresponds to areas based on kernel size and dilation rate. For large objects, the DRB effectively extends the receptive field by combining large-kernel convolution layers with dilated convolution layers. For small objects, the dilated convolution layers in the DRB, using small kernels and varying dilation rates, efficiently capture fine-grained features, thereby improving multi-scale vehicle detection performance in complex backgrounds.

3.3. Soft-NMS

In the process of vehicle detection and tracking, Soft-NMS is used to replace traditional NMS to improve performance in densely populated vehicle scenarios. The traditional NMS steps are as follows:

1. Sort the candidate boxes in descending order based on their confidence scores.
2. Select the candidate box with the highest confidence score and add it to the final result set.
3. Compute the IoU between this candidate box and the other boxes. If a candidate box's IoU exceeds a set threshold, it is deleted.
4. From the remaining candidate boxes, select the one with the highest confidence score, and repeat steps 2 and 3 until all candidate boxes have been processed.

Soft-NMS optimizes the third step of traditional NMS: in Soft-NMS, after calculating the IoU between the candidate box with the highest confidence score and the other boxes, instead of deleting those boxes with an IoU exceeding the threshold, a soft handling method is used by decaying their confidence scores. Specifically, we adopt a Gaussian decay method, as shown in Equation (1):

$$s_i = s_i \times \exp\left(-\frac{\text{IoU}^2}{\sigma}\right) \quad (1)$$

where s_i represents the confidence score of the i -th candidate box, and σ is the decay factor.

If the decayed confidence score is below the set threshold, the candidate box will be removed. In scenarios with dense vehicles, the overlapping of candidate boxes between targets is severe. Compared to traditional NMS, Soft-NMS can retain more valid candidate boxes, handle adjacent targets more effectively, and reduce the occurrence of missed detections.

3.4. Improved Kalman Filter

In two-stage tracking algorithms, the output of the bounding box is usually estimated using the Kalman filter (KF) to determine the target's motion state. In ByteTrack, the state vector of the Kalman filter is shown in Equation (2):

$$\mathbf{x} = [x_c, y_c, a, h, \dot{x}_c, \dot{y}_c, \dot{a}, \dot{h}]^T \quad (2)$$

where (x_c, y_c) are the center coordinates, a is the aspect ratio, and h is the height.

Then, when dealing with partial occlusion and perspective changes, the aspect ratio is difficult to adapt. Therefore, directly estimating the width and height is a better choice. The improved state vector is shown in Equation (3):

$$\mathbf{x}_k = [x_c(k), y_c(k), w(k), h(k), \dot{x}_c(k), \dot{y}_c(k), \dot{w}(k), \dot{h}(k)] \quad (3)$$

where $(x_c(k), y_c(k))$ are the center coordinates, $w(k)$ is the width, and $h(k)$ is the height.

Compared to using the aspect ratio, directly estimating the width and height has the following advantages: on one hand, it can more accurately reflect the actual size of the object and better adapt to changes in vehicle dimensions; on the other hand, changes in the shape and size of the vehicle during motion are complex, and using the aspect ratio increases nonlinear errors. Directly estimating the width and height can reduce the distortion of the bounding box caused by the accumulation of errors, thus maintaining better stability in complex scenes.

In the classical Kalman filter, the process noise covariance matrix Q and the measurement noise covariance matrix R are set as fixed values, which remain unchanged throughout the tracking process. To enhance the performance of the Kalman filter in dynamic environments, Q and R are dynamically adjusted according to real-time conditions, Q as shown in Equation (4) and R as shown in Equation (5).

$$Q_k = \text{diag} \left(\begin{aligned} & \left(\sigma_p \hat{w}_{k-1|k-1} \right)^2, \left(\sigma_p \hat{h}_{k-1|k-1} \right)^2, \\ & \left(\sigma_p \hat{w}_{k-1|k-1} \right)^2, \left(\sigma_p \hat{h}_{k-1|k-1} \right)^2, \\ & \left(\sigma_v \hat{w}_{k-1|k-1} \right)^2, \left(\sigma_v \hat{h}_{k-1|k-1} \right)^2, \\ & \left(\sigma_v \hat{w}_{k-1|k-1} \right)^2, \left(\sigma_v \hat{h}_{k-1|k-1} \right)^2 \end{aligned} \right) \quad (4)$$

where σ_p is the standard deviation of position noise and σ_v is the standard deviation of velocity noise.

$$R_k = \text{diag} \left(\begin{aligned} & \left(\sigma_m \hat{w}_{k|k-1} \right)^2, \left(\sigma_m \hat{h}_{k|k-1} \right)^2, \\ & \left(\sigma_m \hat{w}_{k|k-1} \right)^2, \left(\sigma_m \hat{h}_{k|k-1} \right)^2 \end{aligned} \right) \quad (5)$$

where σ_m is the standard deviation of measurement noise.

Time-varying Q and R can more flexibly respond to nonlinear changes in the environment, such as the deceleration or acceleration of a vehicle. Compared to fixed covariance matrices, time-varying Q and R can be adjusted based on the current prediction and measurement states, more effectively addressing issues of noise and error accumulation. This allows for better adaptation to dynamic environments and improves the overall performance of the system.

3.5. Gaussian Smoothed Interpolation

To address the issue of trajectory loss caused by detection omissions in vehicle tracking, the Gaussian Smoothed Interpolation method is employed as a post-processing step for optimization. Traditional linear interpolation methods typically rely only on the positions of two endpoints, ignoring the complexity of vehicle motion, which results in less accurate outcomes. GSI captures the nonlinear motion trajectory of the vehicle through Gaussian Process Regression (GPR), thereby providing more precise results.

For the i th vehicle trajectory, the GSI model is represented by Equation (6):

$$p_t = f^{(i)}(t) + \epsilon \quad (6)$$

where t is the frame ID, p_t is the vehicle position coordinate at frame t , $f^{(i)}(t)$ is the function fitted by GPR used to describe the vehicle trajectory, and $\epsilon \sim N(0, \sigma^2)$ represents Gaussian noise.

The Gaussian Process is a powerful non-parametric Bayesian method that can capture the nonlinear relationships between data points, assuming the function $f^{(i)}$ follows a Gaussian Process, as shown in Equation (7):

$$f^{(i)} \sim GP(0, k(x, x')) \quad (7)$$

where the covariance function $k(x, x')$ is the Radial Basis Function (RBF) kernel, which is particularly sensitive to local changes. The closer the data points are, the higher their correlation, and the response of the RBF kernel quickly diminishes as the distance between data points increases. This characteristic makes the RBF kernel more suitable for describing nonlinear changes in vehicle motion, such as starting, turning, and accelerating or decelerating. Specifically represented as shown in Equation (8):

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\lambda^2}\right) \quad (8)$$

The process of trajectory position prediction and interpolation is as follows: for frames with missing detections, the smoothed position P^* is predicted as shown in Equation (9):

$$P^* = K(F^*, F) \left(K(F, F) + \sigma^2 I \right)^{-1} P \quad (9)$$

where $K(\cdot, \cdot)$ is the covariance matrix based on the covariance function $k(\cdot, \cdot)$, I is the identity matrix, and P represents the known vehicle positions of the trajectory.

In addition, for the RBF kernel function, the scale parameter λ determines its sensitivity to data changes. During vehicle tracking, different vehicle trajectory lengths may correspond to different motion patterns. Therefore, by adjusting λ to accommodate the length l of the vehicle trajectory, the trajectories can be smoothed, thereby reducing noise and achieving smoother and more accurate tracking. The setting of λ is shown in Equation (10):

$$\lambda = \tau \cdot \log\left(\frac{\tau^3}{l}\right) \quad (10)$$

where τ is a constant used to control the smoothing degree, which has been set to 18 based on experimental results obtained through grid search.

Through the above steps, GSI can effectively fill in missing trajectories in vehicle tracking, improving the accuracy and stability of trajectory predictions.

4. Experiments and Results

4.1. Dataset and Experimental Configuration

The UA-DETRAC dataset [34], jointly released by the University of Alberta and the University of Science and Technology of China, serves as a benchmark dataset specifically designed for multi-object vehicle detection and tracking. This dataset comprises 10 h of video captured using a Canon EOS 550D camera (Canon USA, Inc., Lake Success, NY, USA). The videos are recorded at a frame rate of 25 frames per second, encompassing over 140,000 frames and 1.21 million annotated object detection boxes. The training set consists of 60 video sequences, totaling 83,791 images, while the test set includes 40 video sequences, totaling 56,340 images. The UA-DETRAC dataset presents various challenges, with representative scenarios illustrated in Figure 5. The dataset covers four vehicle categories: cars, buses, vans, and others. Additionally, it includes four weather conditions: sunny, night, cloudy, and rainy, as well as scenes with high vehicle density and occlusion rates.



Figure 5. Representative scenarios in the UA-DETRAC dataset.

In the vehicle detection experiments, images without labels were first excluded, resulting in a training set of 82,085 images and a test set of 56,167 images. To avoid overfitting, every 10th frame was sampled, ultimately obtaining 8209 images for training and 5617 images for evaluation. For the vehicle tracking experiments, 40 video sequences from the test set were used for evaluation.

The GPU device utilized in these experiments is an NVIDIA GeForce RTX 4090 (NVIDIA Corporation, Santa Clara, CA, USA), and the CPU device is a Core i9-9900K @ 3.60 GHz CPU (Intel Corporation, Chengdu, China). The experiments were conducted on a Windows 10 system with a deep learning environment based on PyTorch 1.31.1, CUDA 11.6, and Python 3.8. The training parameters were set as follows: the number of training epochs was 200, the batch size was 32, and no pre-trained weights were used during the training process.

4.2. Evaluation Metrics

In vehicle detection, Precision (P), Recall (R), mAP@0.5, and mAP@0.5:0.95 are used to evaluate the experimental results.

P: the proportion of true-positive samples among all samples detected as positive, as shown in Equation (11):

$$P = \frac{TP}{TP + FP} \quad (11)$$

R: the proportion of correctly detected positive samples among all actual positive samples, as shown in Equation (12):

$$R = \frac{TP}{TP + FN} \quad (12)$$

where TP is the number of true-positive samples, FP is the number of false-positive samples, and FN is the number of actual positive samples that were incorrectly detected as negative samples.

Average Precision (AP): the area under the P-R curve for a certain class of samples, as shown in Equation (13):

$$AP = \int_0^1 P dR \quad (13)$$

mAP@0.5 is the mean Average Precision for all classes with an IoU threshold of 0.5.

mAP@0.5:0.95 is the mean Average Precision for all classes with IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05.

In vehicle tracking, MOTA, IDF1, and IDS are used to evaluate the experimental results.

MOTA measures the accuracy of the tracking algorithm, considering various errors that occur during the tracking process, such as false positives, missed detections, and identity switches. This is shown in Equation (14):

$$\text{MOTA} = 1 - \frac{\sum_t (\text{FN}_t + \text{FP}_t + \text{IDSW}_t)}{\sum_i \text{GT}_t} \quad (14)$$

where FN_t represents the number of missed detections at time t , FP_t represents the number of false positives at time t , IDSW_t represents the number of identity switches at time t , and GT_t represents the number of ground truth targets at time t .

IDF1: IDF1 measures the identity consistency and continuity of the tracking algorithm, as shown in Equation (15):

$$\text{IDF1} = 2 \cdot \frac{\text{IDP} \cdot \text{IDR}}{\text{IDP} + \text{IDR}} \quad (15)$$

where IDP represents the proportion of correctly identified targets among the detected targets. IDR represents the proportion of correctly detected target identities to the actual target identities.

IDS: The number of identity switches that occur during the tracking process, meaning the same target is assigned different identities. A lower value indicates better performance of the tracker in maintaining identity consistency.

4.3. Detector Ablation Experiments

To verify the effectiveness of the proposed improvement modules in vehicle detection, ablation experiments were conducted using YOLOv8-S as the baseline model. The results are shown in Table 1:

Table 1. Ablation experiment results on the UA-DETRAC dataset.

Model	P	R	mAP@0.5	mAP@0.5:0.95	Flops
YOLOv8-S	71.8%	58.6%	64.2%	46.6%	28.8 G
YOLOv8-S+CG_Down	78.4%	63.3%	69.9%	51.0%	30.7 G
YOLOv8-S+C2f_DRB	81.5%	62.6%	71.4%	51.6%	24.7 G
YOLOv8-S+CG_Down+C2f_DRB	80.7%	63.9%	72.9%	52.2%	26.6 G
YOLOv8-S+CG_Down+C2f_DRB+Soft-NMS	82.4%	62.3%	73.2%	55.4%	26.6 G

As seen from the data in Table 1, the introduction of the CG module in the down-sampling process increased P by 6.6%, R by 4.7%, mAP@0.5 by 5.7%, and mAP@0.5:0.95 by 4.4%, verifying the effectiveness of the CG module in extracting vehicle features. The introduction of the C2f_DRB module increased P by 9.7%, R by 4.0%, mAP@0.5 by 7.2%, and mAP@0.5:0.95 by 5%, demonstrating the significant advantage of C2f_DRB in handling multi-scale issues. When both the CG and C2f_DRB modules were introduced simultaneously, P increased by 8.9%, R by 5.3%, mAP@0.5 by 8.7%, and mAP@0.5:0.95 by 5.6%, proving that the combination of these two modules can better extract and integrate vehicle features. Finally, replacing traditional NMS with Soft-NMS for post-processing slightly increased mAP@0.5 by 0.3 percentage points, but significantly improved mAP@0.5:0.95 by 3.2%, laying a foundation for more precise vehicle tracking in subsequent stages. Compared to the baseline model, with a reduction of 2.2 G in Flops, P increased by 10.6%, R by 3.7%, mAP@0.5 by 9%, and mAP@0.5:0.95 by 8.8%, fully verifying the superiority of CDS-YOLOv8 in handling vehicle detection scenarios.

To further demonstrate the advancements of the improved model in vehicle detection, a visual analysis of the experimental results is presented, as shown in Figure 6. The images from left to right represent the original image, the prediction results of the baseline model YOLOv8-S, the prediction results of the improved model CDS-YOLOv8, and the adjustments made by CDS-YOLOv8 compared to YOLOv8-S, which include the reduction in missed vehicles. The optimized parts are marked with pink circles. Additionally, the black areas in the images are ignored regions, as the subsequent vehicle tracking only focuses on moving vehicles on the main road.



Figure 6. Visualization of Detection Results.

In Figure 6, we selected three representative scenarios: sunny, nighttime, and rainy. In the sunny scenario, as shown in the pink circled area, CDS-YOLOv8 detected a bus that YOLOv8 missed, as well as cars farther from the camera. In the nighttime scenario, CDS-YOLOv8 detected almost all the targets that YOLOv8 missed and accurately identified a partially occluded white car that YOLOv8 did not recognize as a valid target. In the rainy scenario, CDS-YOLOv8 detected almost all the small targets that were far from the camera and identified a taxi that were partially occluded and not recognized as valid targets by YOLOv8.

In these scenarios, CDS-YOLOv8 demonstrated significant improvements over YOLOv8-S in feature extraction capabilities and multi-scale processing. Additionally, it retained more valid targets in dense vehicle scenes by handling highly overlapping candidate boxes using Soft-NMS with confidence decay rather than simply eliminating them through an IoU threshold. In summary, CDS-YOLOv8 achieved excellent performance under various lighting and weather conditions, effectively handling multi-scale issues and dense vehicle scenarios.

4.4. Comparison Experiment of Detectors

To demonstrate the advancement of CDS-YOLOv8 in vehicle detection, we compared it with mainstream detection algorithms. The results are shown in Table 2:

Table 2. Comparative Experimental Results on the UA-DETRAC Dataset. (The best results are highlighted in bold.)

Model	mAP@0.5	mAP@0.5:0.95	Flops
Faster-RCNN [20]	66.3%	38.8%	60.5 G
Cascade-RCNN [21]	65.2%	41.1%	88.1 G
RetinaNet [19]	56.6%	32.1%	45.2 G
ATSS [35]	58.8%	32.1%	49.2 G
YOLOv5-S [36]	62.0%	44.9%	16 G
YOLOv6-S [37]	59.0%	43.0%	44.9 G
YOLOv7 [38]	71.1%	53.9%	105.2 G
YOLOv8-S [13]	64.2%	46.6%	28.8 G
YOLOv8-M [13]	68.2%	50.6%	79.3 G
YOLOv8-L [13]	71.6%	51.4%	165.7 G
YOLOv9-S [39]	65.1%	48.0%	39.6 G
L. Kang et al. [23]	70.0%	50.3%	26.7 G
CDS-YOLOv8 (ours)	73.2%	55.4%	26.6 G

From the comparison results, it is evident that CDS-YOLOv8 exhibits the most advanced performance compared to these mainstream two-stage and one-stage detection methods. Its

performance not only surpasses the latest version of YOLOv9-S but also outperforms the large version of YOLOv8, achieving improvements of 1.6% and 4% in mAP@0.5 and mAP@0.5:0.95, respectively. CDS-YOLOv8 strikes a better balance between accuracy and computational complexity, further proving its superiority in vehicle detection scenarios.

4.5. Tracker Ablation Experiment

To verify the effectiveness of the improved detector and the proposed improvement modules in vehicle tracking, an ablation experiment was conducted as shown in Table 3. The experimental results are the averages based on 40 test sequences:

Table 3. Ablation Experiment Results on the UA-DETRAC Dataset.

Detector+ Tracker	mIDF1	IDS	mMOTA
YOLOv8s+ BYTETrack	76.9%	855	67.2%
CDS-YOLOv8+ BYTETrack	78.2%	785	70.6%
CDS-YOLOv8+ BYTETrack+ Improved KF	79.7%	717	73.1%
CDS-YOLOv8+ BYTETrack+ Improved KF +GSI	80.3%	530	73.9%

Compared to the original detector, the introduction of the improved detector CDS-YOLOv8 increases mIDF1 by 1.3%, reduces IDS by 70, and significantly enhances mMOTA by 3.4%. This reflects the critical role of the detector in the tracking system. Subsequently, after improving the tracker by optimizing the Kalman filter, mIDF1 increases by 1.5%, IDS decreases by 68, and mMOTA improves by 2.5%. This indicates that the improved Kalman filter can more flexibly handle the nonlinear motion of vehicles. On this basis, introducing GSI to process the tracking trajectories further increases mIDF1 by 0.6%, significantly reduces IDS by 187, and enhances mMOTA by 0.8%, demonstrating the effectiveness of this method in filling trajectory gaps. Using the improved detector and tracker, compared to before the improvements, mIDF1 increases by 3.4%, IDS decreases by 325, and mMOTA improves by 6.7%, significantly enhancing vehicle tracking performance.

To more intuitively demonstrate the effectiveness of the improved detector and tracker in vehicle tracking, a visualization analysis of the tracking results was performed, as shown in Figure 7. The four columns of images from left to right are the original image, the tracking results using the original detector and tracker, the tracking results using the improved detector and tracker, and the optimization effects of the improved model compared to the original detector and tracker.



Figure 7. Visualization of Tracking Results.

In Figure 7, we also selected three representative scenarios: sunny, nighttime, and rainy. In the sunny scenario, as shown in the pink circled area, CDS-YOLOv8—compared to YOLOv8—not only detected more missed targets but also provided more accurate bounding box detection. For example, CDS-YOLOv8 was able to more comprehensively identify a white van and a taxi at the edge of the image, whereas YOLOv8 only identified a partial

region as a bounding box. Additionally, the tracking ID of the white van was 1281 with the original detector and tracker but it significantly dropped to 1099 after improvements, indicating a notable enhancement in tracking consistency. In the nighttime scenario, the improved detector identified more small targets, and the tracking ID of the bus on the left decreased significantly from 3060 to 2529. In the rainy scenario, YOLOv8 identified the front of a police car as a separate vehicle, while CDS-YOLOv8 correctly recognized the entire bounding box of the police car. Furthermore, the tracking ID of the police car significantly decreased from 5898 to 5026 after the improvements.

Overall, the improved detector not only reduced the number of missed detections but also provided more accurate bounding box detections. The significant reduction in ID numbers for the same target indicates that the improved tracker maintained high accuracy and consistency in vehicle tracking. The enhanced detector and tracker are better equipped to handle complex scenarios and nonlinear changes in vehicles in urban environments.

4.6. Comparison Experiments of Trackers

To validate the advancement of the improved tracker, comparative experiments were conducted with mainstream two-stage object tracking algorithms. To ensure fairness, all experiments used CDS-YOLOv8 for the detector. The tracking results are shown in Table 4:

Table 4. Comparison Results on UA-DETRAC Dataset. (Best results are highlighted in bold.)

Tracker	mIDF1	IDS	mMOTA
DeepSORT [29]	73.5%	2731	70.0%
StrongSORT [17]	76.8%	2994	71.3%
OCSORT [40]	76.8%	2191	71.9%
DeepOCSORT [41]	77.1%	1830	72.1%
BYTETrack [16]	78.2%	785	70.6%
BOT-SORT [30]	78.6%	727	72.9%
Improved BYTETrack (ours)	80.3%	530	73.9%

From the comparison results, it can be seen that the improved BYTETrack demonstrates state-of-the-art performance. Analyzing the results of BYTETrack and BOT-SORT tracking algorithms, it is evident that considering low-confidence score boxes can significantly reduce IDS, proving its effectiveness in identity consistency. Additionally, DeepSORT, StrongSORT, DeepOCSORT, and BOT-SORT all employ the ReID method, which greatly reduces the network's operating speed. In contrast, the improved BYTETrack presented in this paper does not use the ReID method and still achieves the highest performance across all three tracking metrics, achieving a good balance between tracking speed and accuracy.

5. Conclusions

This paper addresses the challenges encountered in vehicle detection and tracking, such as complex scenarios, scale diversity caused by varying vehicle sizes and camera angles, and partial occlusions in dense scenes. A series of improvements were proposed to tackle these issues.

In terms of vehicle detection, we introduce the CG module during the downsampling process, which combines local and surrounding features to form joint features. These joint features are then weighted using global information, enhancing the feature extraction capability of the detector. Additionally, the DRB module was reconstructed and integrated into the C2f module. Using large-kernel convolution layers and multiple small-kernel convolutions with dilations, different scales and sparse features were extracted. Reparameterization technology was employed to generate equivalent convolution kernels, allowing for better comprehensive feature extraction during the inference stage, effectively addressing multi-scale problems. We also replaced the traditional NMS with Soft-NMS, which better handles partial occlusion scenarios through smooth confidence decay. In the aspect of vehicle tracking, we improved the state vector of the Kalman filter and introduced

a time-related covariance matrix to better cope with dynamic vehicle movements. Additionally, the GSI method was incorporated to fill in trajectory gaps caused by missed detections. We conducted experiments on the UA-DETRAC dataset, using ablation studies to demonstrate the role of each module and employing visual analysis to intuitively show the optimization effects of the improved model. Comparative experiments showcased the advanced performance of the CDS-YOLOv8 and the improved BYTETrack algorithm.

In the future, we will focus more on vehicle detection and tracking from different perspectives, while exploring more diverse modalities, such as infrared images, radar, and other methods, to enhance the effectiveness of vehicle detection and tracking.

Author Contributions: Conceptualization, L.Y.; methodology, L.Y.; software, L.Y.; validation, L.Y. and Y.C.; formal analysis, C.S.; investigation, C.S. and R.L.; data curation, L.Y. and C.X.; writing—review and editing, L.Y.; visualization, L.Y.; supervision, Y.C.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Talent Fund project of China West Normal University (No. 463177).

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the first author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Boukerche, A.; Tao, Y.; Sun, P. Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems. *Comput. Netw.* **2020**, *182*, 107484. [CrossRef]
2. Ju, Y.; Cao, Z.; Chen, Y.; Liu, L.; Pei, Q.; Mumtaz, S.; Dong, M.; Guizani, M. NOMA-Assisted Secure Offloading for Vehicular Edge Computing Networks with Asynchronous Deep Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 2627–2640. [CrossRef]
3. Roy, D.; Li, Y.; Jian, T.; Tian, P.; Chowdhury, K.; Ioannidis, S. Multi-Modality Sensing and Data Fusion for Multi-Vehicle Detection. *IEEE Trans. Multimed.* **2023**, *25*, 2280–2295. [CrossRef]
4. Humayun, M.; Ashfaq, F.; Jhanjhi, N.Z.; Alsadun, M.K. Traffic Management: Multi-Scale Vehicle Detection in Varying Weather Conditions Using YOLOv4 and Spatial Pyramid Pooling Network. *Electronics* **2022**, *11*, 2748. [CrossRef]
5. Deshmukh, P.; Satyanarayana, G.S.R.; Majhi, S.; Sahoo, U.K.; Das, S.K. Swin Transformer Based Vehicle Detection in Undisciplined Traffic Environment. *Expert Syst. Appl.* **2023**, *213*, 118992. [CrossRef]
6. Wang, J.; Dong, Y.; Zhao, S.; Zhang, Z. A High-Precision Vehicle Detection and Tracking Method Based on the Attention Mechanism. *Sensors* **2023**, *23*, 724. [CrossRef] [PubMed]
7. Bui, T.; Wang, G.; Wei, G.; Zeng, Q. Vehicle Multi-Object Detection and Tracking Algorithm Based on Improved You Only Look Once 5s Version and DeepSORT. *Appl. Sci.* **2024**, *14*, 2690. [CrossRef]
8. Wu, Y.; Wang, T.; Gu, R.; Liu, C.; Xu, B. Nighttime vehicle detection algorithm based on image translation technology 1. *J. Intell. Fuzzy Syst.* **2024**, *46*, 5377–5389. [CrossRef]
9. Xu, B.; Liu, X.; Feng, G.; Liu, C. A Monocular-Based Framework for Accurate Identification of Spatial-Temporal Distribution of Vehicle Wheel Loads under Occlusion Scenarios. *Eng. Appl. Artif. Intell.* **2024**, *133*, 107972. [CrossRef]
10. Xu, B.; Liu, C. Keypoint Detection-Based and Multi-Deep Learning Model Integrated Method for Identifying Vehicle Axle Load Spatial-Temporal Distribution. *Adv. Eng. Inform.* **2024**, *62*, 102688. [CrossRef]
11. Xu, H.; Lai, S.; Li, X.; Yang, Y. Cross-Domain Car Detection Model with Integrated Convolutional Block Attention Mechanism. *Image Vis. Comput.* **2023**, *140*, 104834. [CrossRef]
12. Zhang, C.; Yang, C.; Wu, D.; Dong, H.; Deng, B. Cross-View Vehicle Re-Identification Based on Graph Matching. *Appl. Intell.* **2022**, *52*, 14799–14810. [CrossRef]
13. Ultralytics. YOLOv8: Real-Time Object Detection and Image Segmentation. GitHub Repository. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 28 July 2024).
14. Wu, T.; Tang, S.; Zhang, R.; Cao, J.; Zhang, Y. CGNet: A Light-Weight Context Guided Network for Semantic Segmentation. *IEEE Trans. Image Process.* **2021**, *30*, 1169–1179. [CrossRef] [PubMed]
15. Ding, X.; Zhang, Y.; Ge, Y.; Zhao, S.; Song, L.; Yue, X.; Shan, Y. UniRepLKNNet: A Universal Perception Large-Kernel ConvNet for Audio, Video, Point Cloud, Time-Series and Image Recognition. *arXiv* **2024**, arXiv:2311.15599.
16. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. In *Computer Vision—ECCV 2022, Proceedings of the 17th European Conference, Tel Aviv, Israel, 23–27 October 2022*; Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T., Eds.; Springer Nature Switzerland: Cham, Switzerland, 2022; pp. 1–21.

17. Du, Y.; Zhao, Z.; Song, Y.; Zhao, Y.; Su, F.; Gong, T.; Meng, H. StrongSORT: Make DeepSORT Great Again. *IEEE Trans. Multimed.* **2023**, *25*, 8725–8737. [\[CrossRef\]](#)
18. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37.
19. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017*; pp. 2980–2988.
20. Girshick, R. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015*; pp. 1440–1448.
21. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving Into High Quality Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018*; pp. 6154–6162.
22. Chen, Z.; Guo, H.; Yang, J.; Jiao, H.; Feng, Z.; Chen, L.; Gao, T. Fast Vehicle Detection Algorithm in Traffic Scene Based on Improved SSD. *Measurement* **2022**, *201*, 111655. [\[CrossRef\]](#)
23. Kang, L.; Lu, Z.; Meng, L.; Gao, Z. YOLO-FA: Type-1 Fuzzy Attention Based YOLO Detector for Vehicle Detection. *Expert Syst. Appl.* **2024**, *237*, 121209. [\[CrossRef\]](#)
24. Bie, M.; Liu, Y.; Li, G.; Hong, J.; Li, J. Real-Time Vehicle Detection Algorithm Based on a Lightweight You-Only-Look-Once (YOLOv5n-L) Approach. *Expert Syst. Appl.* **2023**, *213*, 119108. [\[CrossRef\]](#)
25. Wang, B.; Li, Y.-Y.; Xu, W.; Wang, H.; Hu, L. Vehicle–Pedestrian Detection Method Based on Improved YOLOv8. *Electronics* **2024**, *13*, 2149. [\[CrossRef\]](#)
26. Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; Wang, S. Towards Real-Time Multi-Object Tracking. In *Computer Vision—ECCV 2020, Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 107–122.
27. Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; Liu, W. FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking. *Int. J. Comput. Vis.* **2021**, *129*, 3069–3087. [\[CrossRef\]](#)
28. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple Online and Realtime Tracking. In *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016*; pp. 3464–3468.
29. Wojke, N.; Bewley, A.; Paulus, D. Simple Online and Realtime Tracking with a Deep Association Metric. In *Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017*; pp. 3645–3649.
30. Aharon, N.; Orfaig, R.; Bobrovsky, B.-Z. BoT-SORT: Robust Associations Multi-Pedestrian Tracking. *arXiv* **2022**, arXiv:2206.14651.
31. Zhao, Z.; Ji, Z.; Yao, Y.; He, Z.; Du, C. Enhanced Detection Model and Joint Scoring Strategy for Multi-Vehicle Tracking. *IEEE Access* **2023**, *11*, 30807–30818. [\[CrossRef\]](#)
32. Zhang, K.; Wu, F.; Sun, H.; Cai, M. Monocular Vehicle Speed Detection Based on Improved YOLOX and DeepSORT. *Neural Comput. Appl.* **2024**, *36*, 9643–9660. [\[CrossRef\]](#)
33. Han, G.; Jin, Q.; Rong, H.; Jin, L.; Zhang, L. Vehicle Tracking Algorithm Based on Deep Learning in Roadside Perspective. *Sustainability* **2023**, *15*, 1950. [\[CrossRef\]](#)
34. Wen, L.; Du, D.; Cai, Z.; Lei, Z.; Chang, M.C.; Qi, H.; Lim, J.; Yang, M.-H.; Lyu, S. UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking. *Comput. Vis. Image Underst.* **2020**, *193*, 102907. [\[CrossRef\]](#)
35. Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; Li, S.Z. Bridging the Gap Between Anchor-Based and Anchor-Free Detection via Adaptive Training Sample Selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020*; pp. 9759–9768.
36. Ultralytics. YOLOv5: Family of Object Detection Architectures and Models Pretrained on the COCO Dataset. GitHub Repository. Available online: <https://github.com/ultralytics/yolov5> (accessed on 28 July 2024).
37. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
38. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023*; pp. 7464–7475.
39. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616.
40. Cao, J.; Pang, J.; Weng, X.; Khirrodar, R.; Kitani, K. Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023*; pp. 9686–9696.
41. Maggiolino, G.; Ahmad, A.; Cao, J.; Kitani, K. Deep OC-Sort: Multi-Pedestrian Tracking by Adaptive Re-Identification. In *Proceedings of the 2023 IEEE International Conference on Image Processing (ICIP), Kuala Lumpur, Malaysia, 8–11 October 2023*; pp. 3025–3029.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Evolution of Antenna Radiation Parameters for Air-to-Plasma Transition

Tomasz Aleksander Miś 

Division of Plasma Physics, Space Research Center of Polish Academy of Sciences, ul. Bartycka 18A, 00-716 Warszawa, Poland; tmis@cbk.waw.pl or tomasz.a.mis@mailplus.pl

Abstract: This paper presents the description of antenna parameters related to its radiation/reception capabilities influenced by the plasma parameters in the environment surrounding the antenna, complementing the existing works on the antenna parameters (e.g., the impedance or currents). The parameters considered are the radiation zones' radiuses (inductive, Fresnel, Fraunhofer), scalloping and directivity; a method of transformation of the air/vacuum-measured radiation/reception pattern to the pattern expected for given plasmatic conditions is also considered. Three different simplified plasma conditions are taken into account (different electron densities: $1.4 \times 10^{12} \text{ m}^{-3}$, $4 \times 10^{11} \text{ m}^{-3}$ and 10^8 m^{-3}), with varying antenna length (1 m, 10 m, 100 m) and signal propagation mode (classic-ionospheric, whistler and Alfvén). The findings show that the presented antenna parameters and its radiation/reception pattern are heavily dependent on the plasma conditions. These findings can be used to form additional requirements and constraints for the mechanical design of new instrumentation for space weather measurements on board spacecraft (e.g., moving the antennas away from the spacecraft in order not to alter their radiation/reception patterns or not to measure the plasma around the spacecraft) or more accurate data processing from existing space weather satellites, allowing, for example, a more precise triangulation of the signal source or its spectral power regarding the actual performance of the antennas submerged in plasma.

Keywords: plasma; plasmasphere; magnetosphere; space antenna; active antenna; antenna parameters; radiation pattern; antenna testing; directivity; scalloping



Citation: Miś, T.A. Evolution of Antenna Radiation Parameters for Air-to-Plasma Transition. *Electronics* **2024**, *13*, 3040. <https://doi.org/10.3390/electronics13153040>

Academic Editors: Jun Wang, Kuikui Fan, Jianfeng Zhu and Fan Wu

Received: 27 June 2024

Revised: 23 July 2024

Accepted: 26 July 2024

Published: 1 August 2024



Copyright: © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction; Electromagnetic Emissions and Antennas in Plasmas

Since the beginning of the space age and the exploration of the near-Earth environment, a high level of interest was put on investigating the properties of the ionized particle sheath that surrounds the planet—the plasmasphere, its variability depending on the solar conditions, its influence on the propagating radio signals and the effect it has or may have on spacecraft and manned space missions [1]. Since the 1960s, numerous naturally occurring radio phenomena have been identified as being produced by the interactions between the solar wind and the Earth's magnetosphere (e.g., the Auroral Kilometric Radiation [2–4] and other types of emissions [5]), many of which are generated within the terrestrial waveguide (the space between the Earth's surface and the lower layers of the ionosphere) and are able to propagate further through the ionosphere, either along the magnetic field lines or independently of them [6–8]. These emissions vary greatly with their propagation paths [9] and present a useful ability to be employed as a remote sensing tool for the ionosphere [10] and the plasmasphere [8], including the magnetopause and the plasmaspheric bow shock [11]. The propagation of these signals is highly dependent on the local plasmatic conditions, both in the ionosphere and the plasmasphere, which has been investigated, e.g., during the occurrence of the X-ray solar flares, significantly affecting both ionized environments [12–15]. It is worth noting that the complex changes in the ionized medium parameters (which do not only include the electron density, but proton density as well) are crucial also for the analysis of the near-planetary environments of the planets exposed to even more intense solar influences, e.g., Mercury [16].

As many of these natural emissions can be received fairly easily on ground [6] and on altitudes closer to the ionosphere [4,17], their propagation through the magnetospheric environment often requires in situ measurements (sometimes in situ only [8]), carried out by dedicated spacecraft and instruments—e.g., Interball-1 [18], POLRAD on board Interball-2 [2], DEMETER [7], SWARM [3], PROGNOZ-8 [19], MAGION-4 [20], ISEE 1 and 2 [5], CESAR [21], CORONAS [21], CLUSTER [22] or Arase [23]. To maximize effective antenna aperture for the reception and, in some cases, active experiments, the space-borne antennas are usually designed as large (up to dozens of meters of total length) dipoles, either trussed or wire-like, with optional supporting for other smaller sensors, e.g., the magnetometers [23,24] or Langmuir probes [25]. While convenient in calculation and testing in the air/vacuum environment, the dipoles however present significantly different parameters and performances when submerged in an ionized environment. Elaborate work has been carried out to determine the loading parameters of such antennas, with respect to the angle of the local magnetic field intensity [26–28] or the current pattern/modes in the antenna wires depending on the local plasma parameters [29,30]. For the active/transmitting antennas, a crucial element is to determine the size of the plasmatic sheath which surrounds the radiator and takes part in the formation of the radio wave [31], as well as to determine the formulas for the electric and magnetic field intensities in 3-dimensional (crucial for the maintenance of accuracy [32]) space around the antenna, both approximated for anisotropic plasma [33], including the spatial distribution of plasma parameters [34] and different types of radio emissions depending on the frequency and propagation mode [35].

These analyses clearly show the significant differences between the air/vacuum cases and the plasma cases for the antenna design and the antenna's basic parameters, such as the input impedance and reactance or the radiation pattern—yet little information is devoted to the actual spatial radiation performance of the antenna, e.g., how does the radiation pattern form and evolve in different plasmatic conditions (this is crucial to determine, e.g., the direction of arrival of the registered signal). These differences are important on both the stage of the antenna design and the antenna operation, which aims to deliver scientifically useful data describing in the most accurate way possible the phenomena appearing inside the plasmasphere. While the more elaborate and accurate formulations for different types of plasmas can be included during the design stage of the antenna, the parameter (e.g., the radiation pattern) verification and testing can be practically carried out in the air conditions only, which demands a simple method of transformation of the in-air results to plasmatic results. What is more, as the actual plasma evolves around the space-borne antenna, the antenna constantly changes its radiation pattern and radiation characteristics (scalloping, directivity, radiation zones)—both in the plasma variables' domain and in the operating frequency domain. The assessment of these changes, even in a simplified yet easily implementable form, is crucial for the analysis of the spectrograms received (e.g., to determine the angle of arrival of the recorded signals or the sensitivity of the antenna system in the given plasmatic conditions in the given frequency range), as well as the assessment of the potential influence of the external objects—booms, satellite structures, other antennas in proximity—on the plasma to which the antenna is subjected (especially including the plasmatic sheath [31]).

Taking all of the considerations above, this paper presents summarized simplified (for isotropic environment, wavenumber as a one-dimensional variable and magnetic field conditions given further) analytic methods for the determination of expected antenna parameters—radiation pattern, scalloping, directivity and spectral sensitivity—for the given plasmatic and geometrical conditions, allowing to better assess the case of antenna operation inside a plasma, facilitating the antenna evaluation during testing and offering an increase in accuracy in the analysis of the registered spectrograms (also spectrograms from the space missions mentioned above)—effectively bridging the gap between the existing antenna analyses (e.g., the impedance and current patterns) and their actual performance in surrounding space.

2. Evolution of the Radiation Zones

The zones of radiation around the antenna can be described as the ranges in which the energy radiated by the antenna is initially (close to the antenna surface) stored in the free space in a reactive form, then—with increasing distance to the antenna—the radio wave front is formed with its basic electric and magnetic field components; the energy takes the radiative form [36]. Figure 1 shows the basic schematic of the zones around a symmetrical dipole, indicating the Fresnel zone radius $r_{Fresnel}$, behind which the reactive form or emitted energy is dominant, and the Fraunhofer zone radius $r_{Fraunhofer}$, after which the radio wave front is fully shaped. Additionally, for a physical antenna dimension h smaller than the wavelength λ , the zone limited by the wavelength (deemed reactive) is also indicated [37].

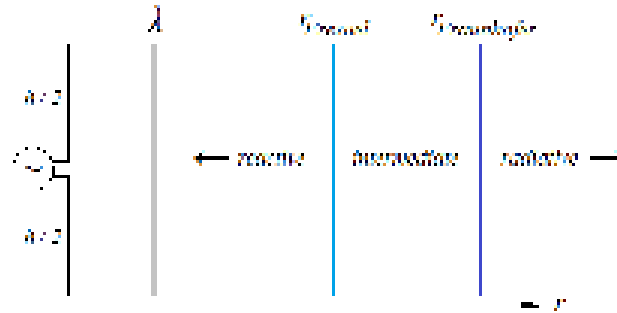


Figure 1. Schematic of the radiation zones around a symmetrical dipole.

The subsequent zone radii for the i -th λ can be defined as [37]:

$$r_{Fresnel} = \lambda_i + \frac{h}{2} \sqrt{\frac{h}{2\lambda_i}} \quad (1)$$

$$r_{Fraunhofer} = \lambda_i + \frac{2h^2}{\lambda_i} \quad (2)$$

For the accurate mathematical formulations of the electric and magnetic field components on elevation θ , orientation ϕ and radial r directions from the antenna, the limiting distance which separates the reactive and radiative zones $r_{Fraunhofer\_approx.}$ can be defined as [38]:

$$r_{Fraunhofer\_approx.} = \left(\frac{4\pi^2}{\lambda_0^2} - \frac{e^2 n_e}{m_e \epsilon_0 c^2} \right)^{-\frac{1}{2}} \quad (3)$$

where λ_0 [m] is the free space wavelength, e is the electron electric charge (1.602×10^{-19} C), n_e [$1/m^3$] is the electron concentration/density, m_e is the electron mass (9.109×10^{-31} kg), ϵ_0 is the free space electrical permittivity (8.854×10^{-12} F/m) and c is the free space light velocity (300,000,000 m/s). For this formulation, the close zone of radiation (reactive) is located at distances $r \ll r_{Fraunhofer\_approx.}$; for the intermediate zone: $r \sim r_{Fraunhofer\_approx.}$ and for the radiative zone: $r \gg r_{Fraunhofer\_approx.}$.

The i -th λ can be calculated as follows:

$$\lambda_i = \frac{2\pi}{k_i} \quad (4)$$

where k_i denoted the wavenumber for a given specific propagation mode. In this paper, three basic propagation modes for lower frequencies are considered—the whistler mode (k_W), i.e., for the frequencies below the local electron plasma frequency, approximated as 1400.564 kHz [35], the Alfvén mode (k_A), i.e., for the frequencies below the local ion gyromagnetic frequency, approximated as 40.215 Hz [35], and the wavenumber k_i calculated for the propagation conditions determined by the ionospheric cutoff frequency (denoted with subscript ‘classic’) for small current-element antenna [38].

For the whistler mode, as given by Bannister et al. [35], the wavenumber is equal to

$$k_W = \frac{1}{c} \sqrt{\frac{2\pi f n_e}{B_0 \epsilon_0}} \quad (5)$$

where f [Hz] is the operating frequency, and B_0 [T] is the magnetic field induction parallel to the direction of propagation of the wave (perpendicular to the antenna axis).

For the Alfvén mode, after Bannister et al. [35], the wavenumber is formulated as follows:

$$k_A = \frac{2\pi f}{c} \sqrt{\frac{n_i m_i}{\epsilon_0}} \quad (6)$$

where n_i [$1/\text{m}^3$] is the ion concentration, equal to n_e , and m_i [kg] is the average ion mass, approximated as 3.17×10^{-26} kg [35].

For the ‘classic’ ionospheric case (in which the wavefront enters the ionosphere and is subjected to absorption, refraction and reflection, solely depending on the frequency and n_e), the wavenumber is given by D. J. Bem [38] as follows:

$$k_J = \sqrt{\frac{4\pi^2}{\lambda_0^2} - \frac{e^2 n_e}{m_e \epsilon_0 c^2}} \quad (7)$$

similarly to the derivation of Formula (3).

The four radiuses of the subsequent radiation zones—inductive (for antenna length h [m] $\ll \lambda_0$), Fresnel, Fraunhofer and Fraunhofer classic—have been calculated for different exemplary electron concentrations: ionospheric F-layer in daylight conditions ($1.4 \times 10^{12} \text{ m}^{-3}$ [38]), ionospheric F-layer for night conditions ($4 \times 10^{11} \text{ m}^{-3}$ [38]) and heliospheric conditions, generalized as 10^8 m^{-3} [39].

The three considered wavenumbers have been plotted in Figures 2–4. Each electron concentration was employed for calculations for three different antenna lengths h : 1 m, 10 m and 100 m (antenna treated as a symmetrical dipole). It can be noticed that the orders of magnitude of the wavenumbers for the Alfvén cases differ significantly from those of the whistler and classic-ionospheric cases—this shall later cause the exclusion of some of the Alfvén cases from direct comparison with other cases due to their gargantuan sizes (when compared to the actual antenna aperture). For the heliospheric case, it can also be noticed that the classic-ionospheric function of k (Formula (7)) starts to overlap with the range for the whistler mode of k function (Formula (5))—it is therefore important to take into consideration the frequency range to which the given function is applicable and the final sizes of the resulting radiation zones’ radiuses, with the priority given to the smaller radius.

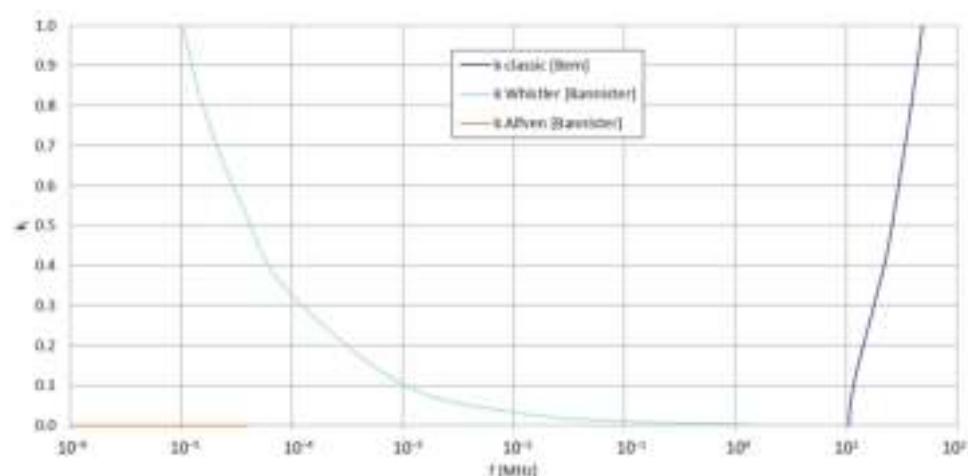


Figure 2. Calculated values for wavenumbers k for the classic (ionospheric) case, whistler propagation mode and Alfvén propagation mode for ionospheric F-layer daylight electron density ($1.4 \times 10^{12} \text{ m}^{-3}$).

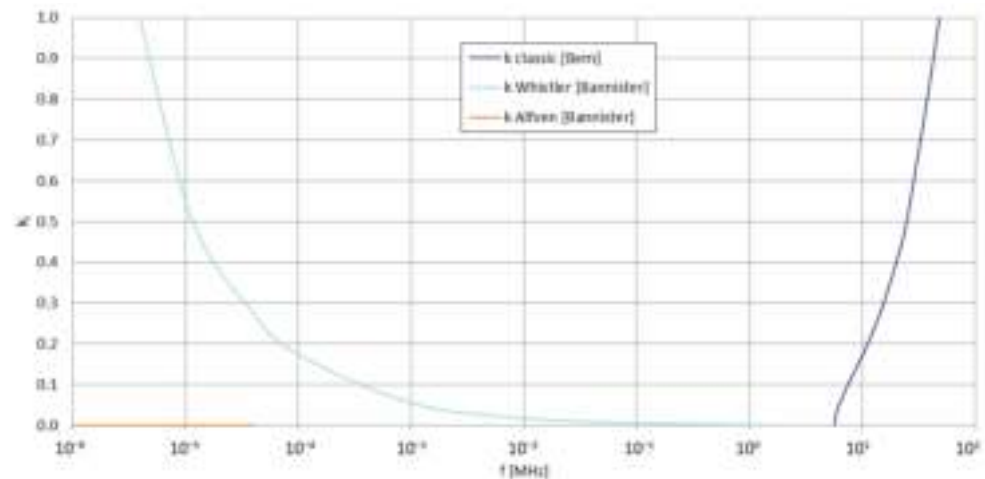


Figure 3. Calculated values for wavenumbers k for the classic (ionospheric) case, whistler propagation mode and Alfvén propagation mode for ionospheric F-layer nighttime electron density ($4 \times 10^{11} \text{ m}^{-3}$).

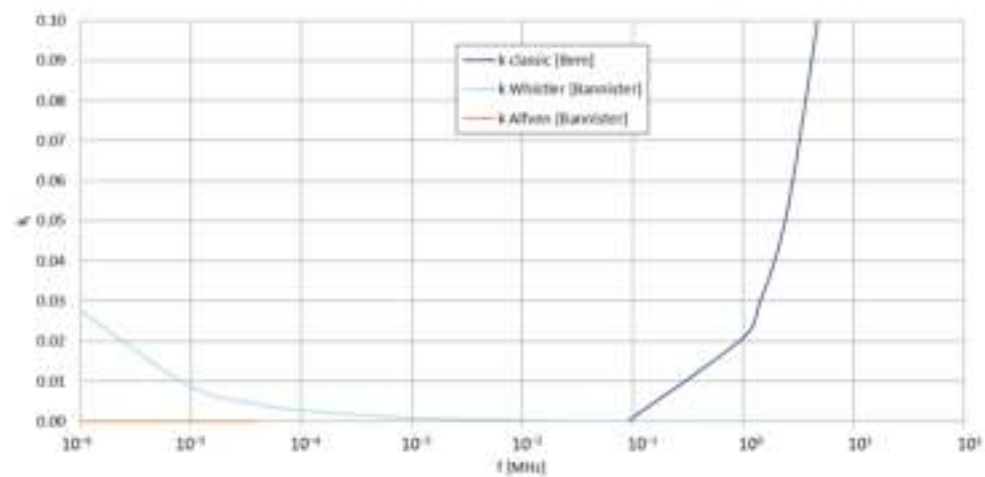


Figure 4. Calculated values for wavenumbers k for the classic (ionospheric) case, whistler propagation mode and Alfvén propagation mode for the approximated heliospheric electron density (10^8 m^{-3}).

Figures 5–7 present the radiation zones' radiuses—Fraunhofer approximate (Formula (3)), Fresnel (Formula (1)), Fraunhofer (Formula (2)) and wavelengths for non-plasmatic cases (for $\lambda \ll h$)—for k_f and $h = 1 \text{ m}$; Figures 8–10: for k_f and $h = 10 \text{ m}$ and Figures 11–13: for k_f and $h = 100 \text{ m}$. It can be noticed that the Fraunhofer zone from Formula (2) expands significantly with the rise of dimensions of the antenna, while the approximate indication of the Fraunhofer zone (e.g., indicating the presence—or lack of presence—of certain fields components) remains more convergent to the inductive range (wavelength), with the Fresnel zone positioned as the largest in these orders of magnitude. These sets of functions and plots shall generally apply for low-MHz radio observations.

Figure 14 shows the approximate Fraunhofer radiation zones' radiuses (3) for k_W and all considered n_e ranges (formula h -independent); Figures 15–17: for the Fresnel zones' radiuses (1) for k_W and all considered n_e ranges and h values; Figures 18–20: for the Fraunhofer zones' radiuses (2) for k_W and all considered n_e ranges and h values; Figures 21–23: for the Fresnel zones' radiuses (1) for k_A and all considered n_e ranges and h values (Fraunhofer zone radiuses for k_A were omitted due to large differences in the orders of magnitudes of their values in comparison to the values of the antenna aperture h).

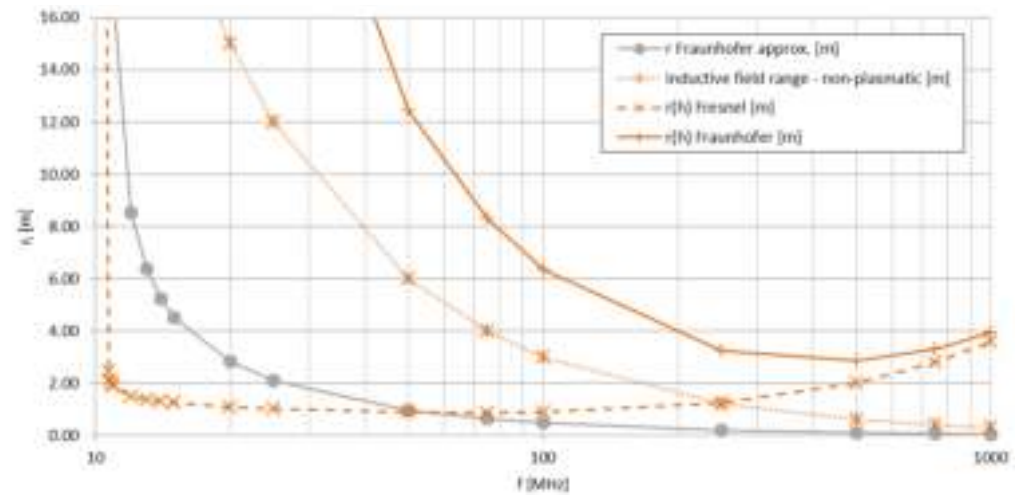


Figure 5. Calculated radiation zones' radiuses for the classic-ionospheric approach (k_f) for ionospheric F-layer daylight electron density ($1.4 \times 10^{12} \text{ m}^{-3}$) and $h = 1 \text{ m}$.

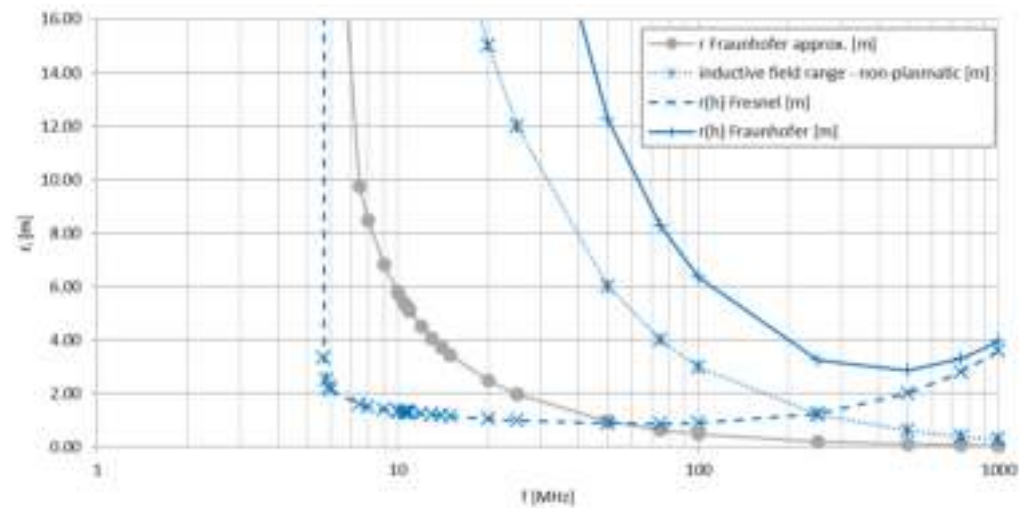


Figure 6. Calculated radiation zones' radiuses for the classic-ionospheric approach (k_f) for ionospheric F-layer nighttime electron density ($4 \times 10^{11} \text{ m}^{-3}$) and $h = 1 \text{ m}$.

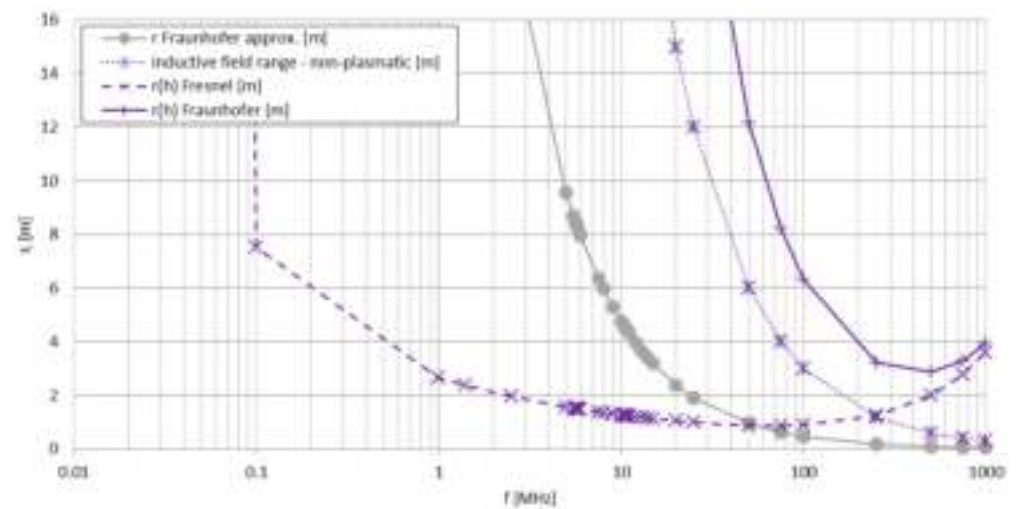


Figure 7. Calculated radiation zones' radiuses for the classic-ionospheric approach (k_f) for the approximated heliospheric electron density (10^8 m^{-3}) and $h = 1 \text{ m}$.

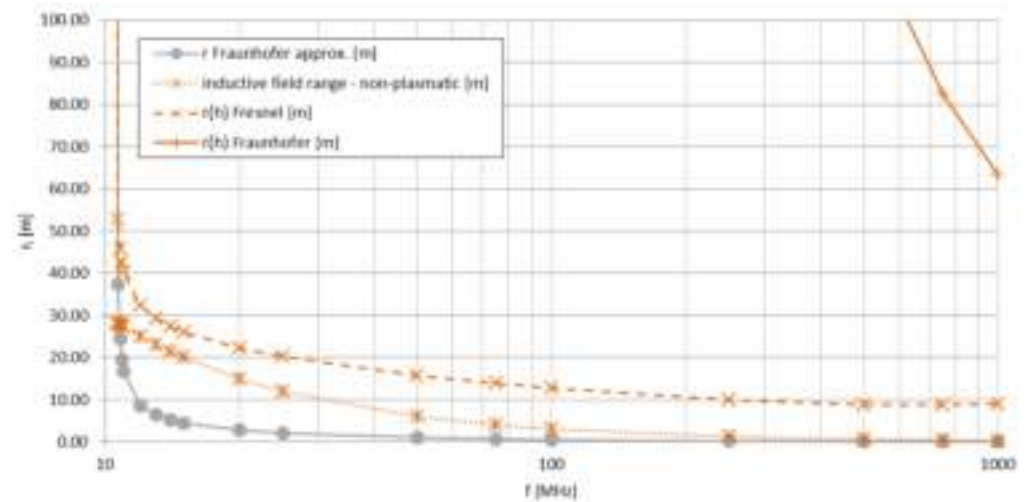


Figure 8. Calculated radiation zones' radiuses for the classic-ionospheric approach (k_f) for ionospheric F-layer daylight electron density ($1.4 \times 10^{12} \text{ m}^{-3}$) and $h = 10 \text{ m}$.

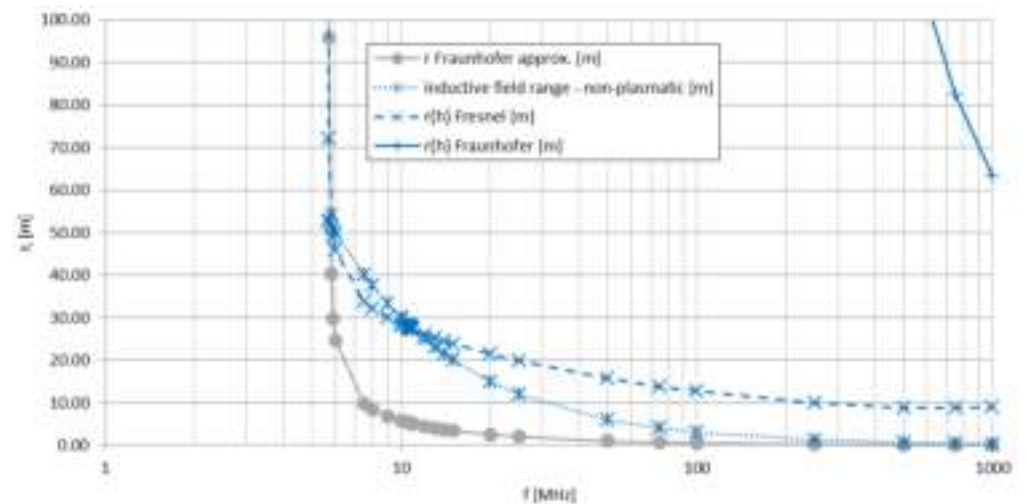


Figure 9. Calculated radiation zones' radiuses for the classic-ionospheric approach (k_f) for ionospheric F-layer nighttime electron density ($4 \times 10^{11} \text{ m}^{-3}$) and $h = 10 \text{ m}$.

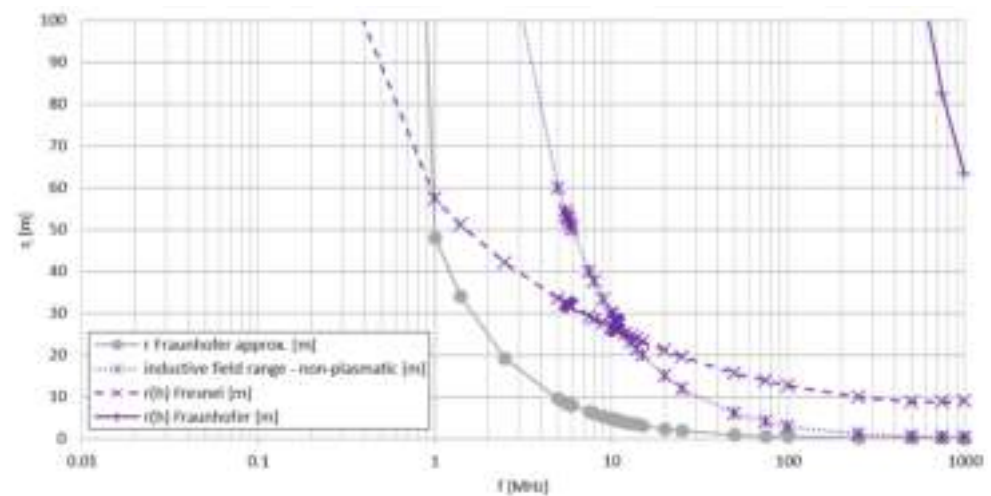


Figure 10. Calculated radiation zones' radiuses for the classic-ionospheric approach (k_f) for the approximated heliospheric electron density (10^8 m^{-3}) and $h = 10 \text{ m}$.

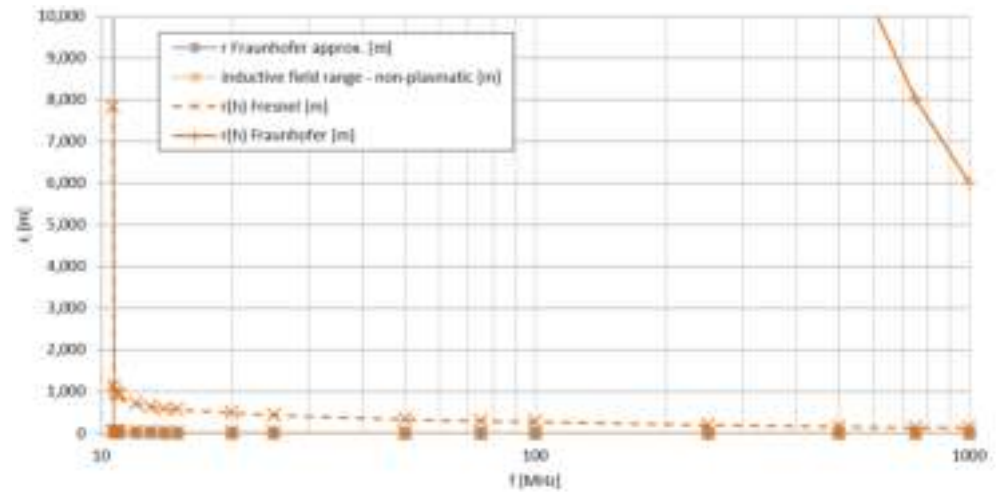


Figure 11. Calculated radiation zones' radii for the classic-ionospheric approach (k_f) for ionospheric F-layer daylight electron density ($1.4 \times 10^{12} \text{ m}^{-3}$) and $h = 100 \text{ m}$.

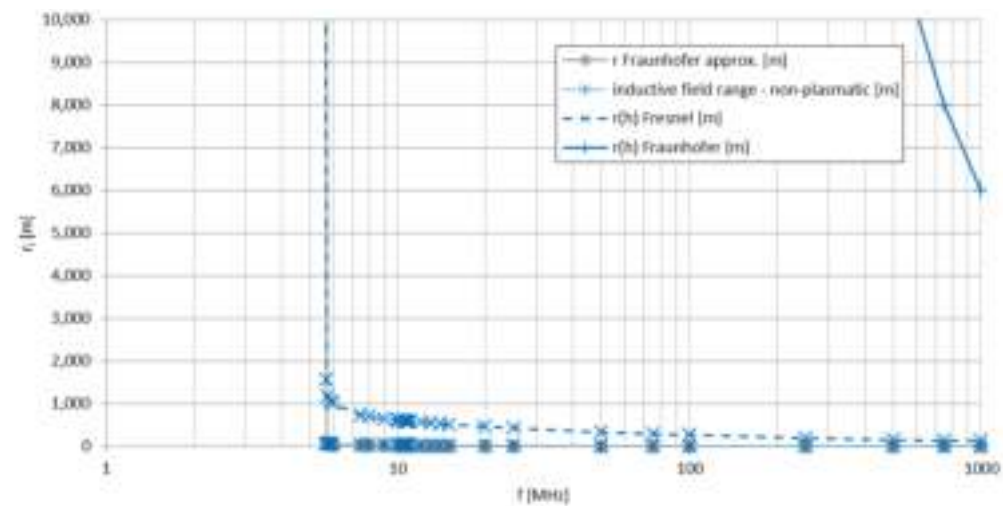


Figure 12. Calculated radiation zones' radii for the classic-ionospheric approach (k_f) for ionospheric F-layer nighttime electron density ($4 \times 10^{11} \text{ m}^{-3}$) and $h = 100 \text{ m}$.

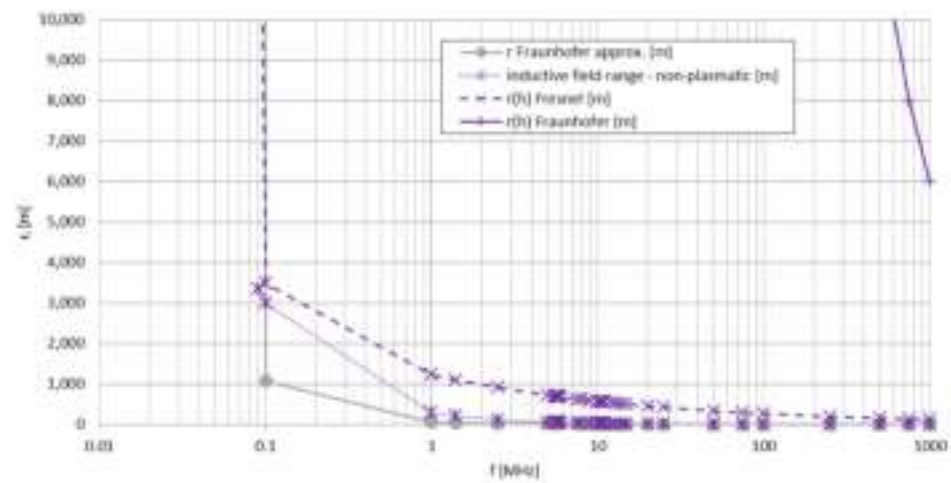


Figure 13. Calculated radiation zones' radii for the classic-ionospheric approach (k_f) for the approximated heliospheric electron density (10^8 m^{-3}) and $h = 100 \text{ m}$.

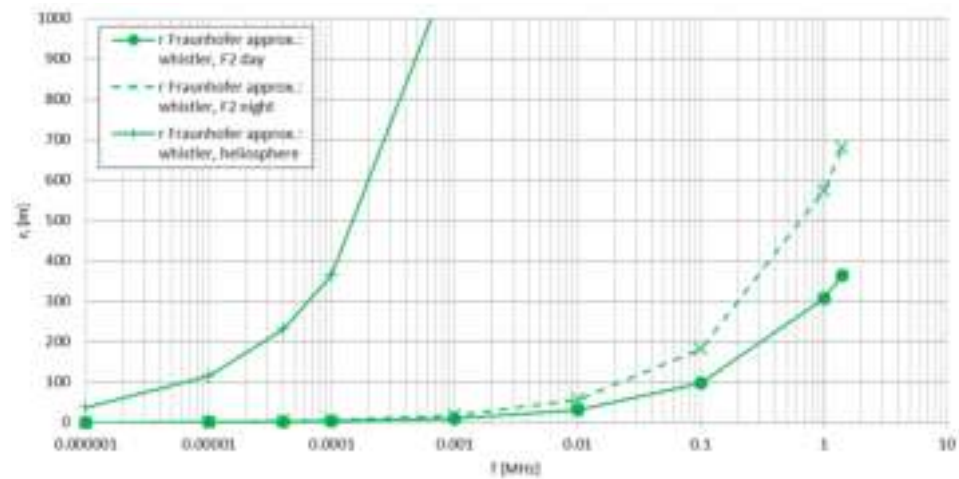


Figure 14. Calculated approximate Fraunhofer radiation zones' radii for the whistler propagation mode (k_W) for $h = 1$ m and different electron concentrations.

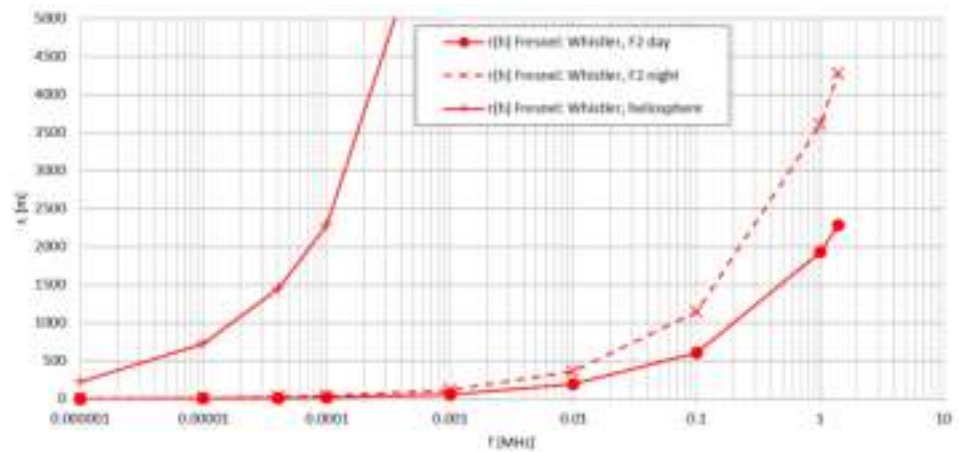


Figure 15. Calculated Fresnel radiation zones' radii for the whistler propagation mode (k_W) for $h = 1$ m and different electron concentrations.

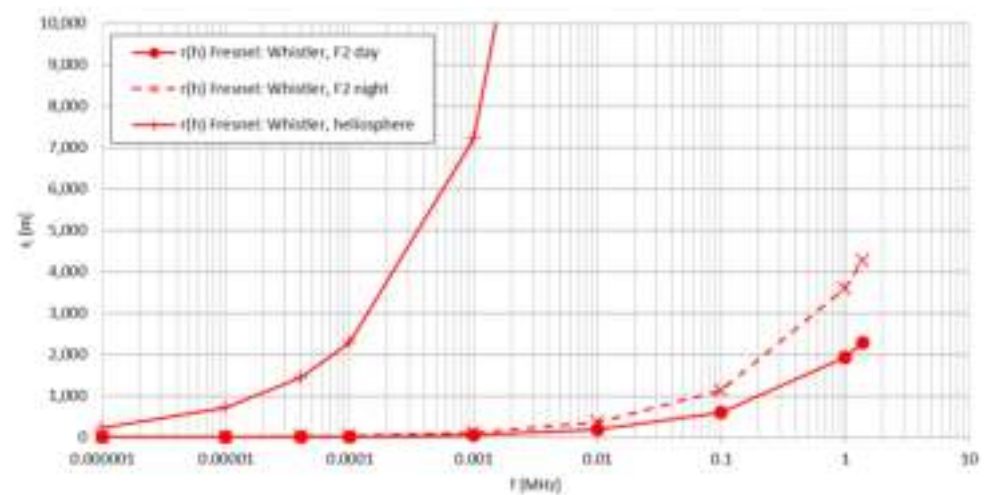


Figure 16. Calculated Fresnel radiation zones' radii for the whistler propagation mode (k_W) for $h = 10$ m and different electron concentrations.

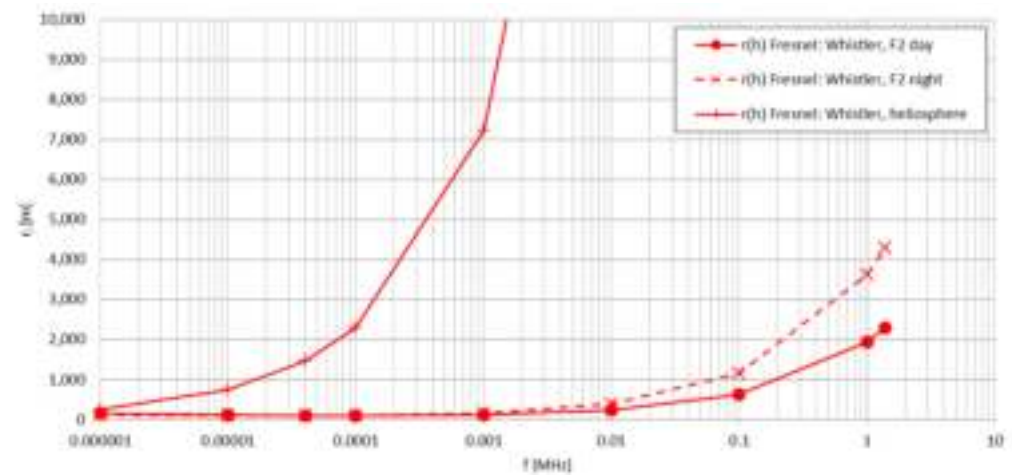


Figure 17. Calculated Fresnel radiation zones' radii for the whistler propagation mode (k_W) for $h = 100$ m and different electron concentrations.

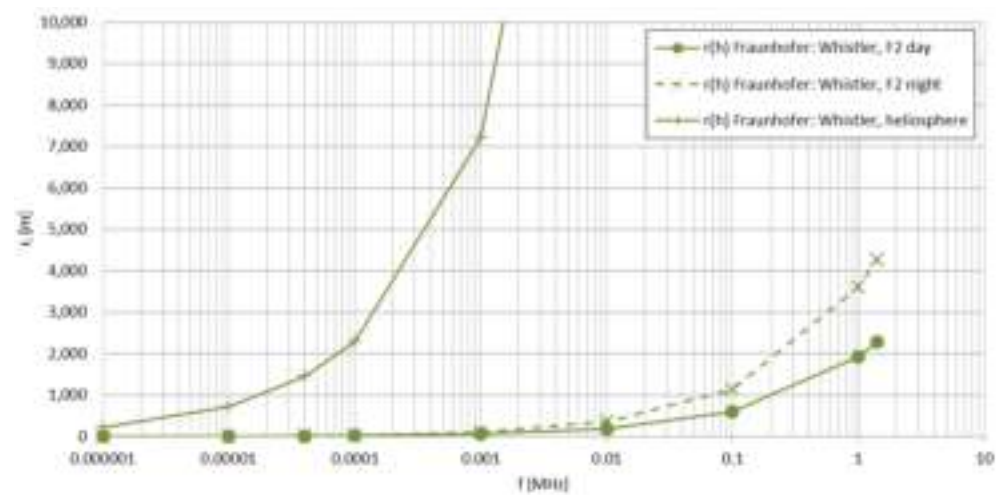


Figure 18. Calculated Fraunhofer radiation zones' radii for the whistler propagation mode (k_W) for $h = 1$ m and different electron concentrations.

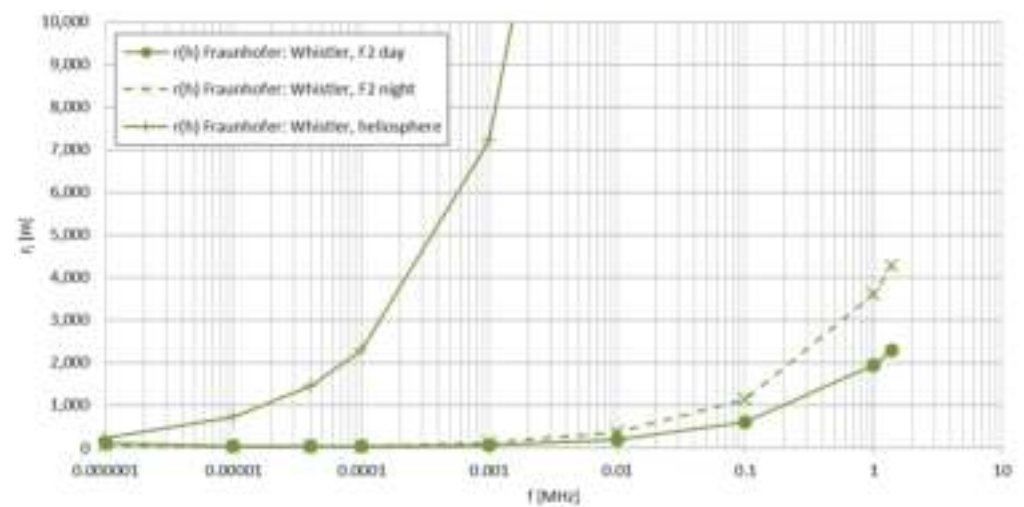


Figure 19. Calculated Fraunhofer radiation zones' radii for the whistler propagation mode (k_W) for $h = 10$ m and different electron concentrations.

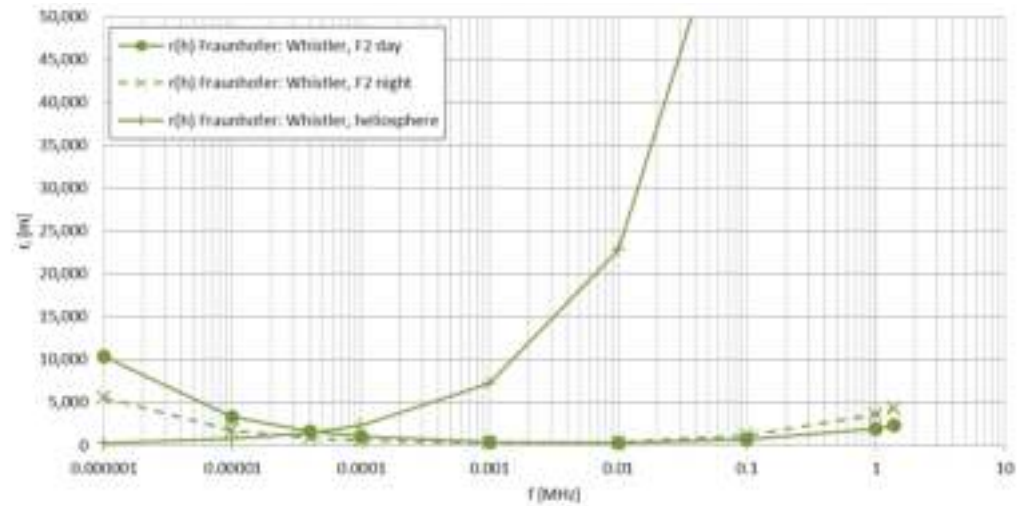


Figure 20. Calculated Fraunhofer radiation zones' radii for the whistler propagation mode (k_W) for $h = 100$ m and different electron concentrations.

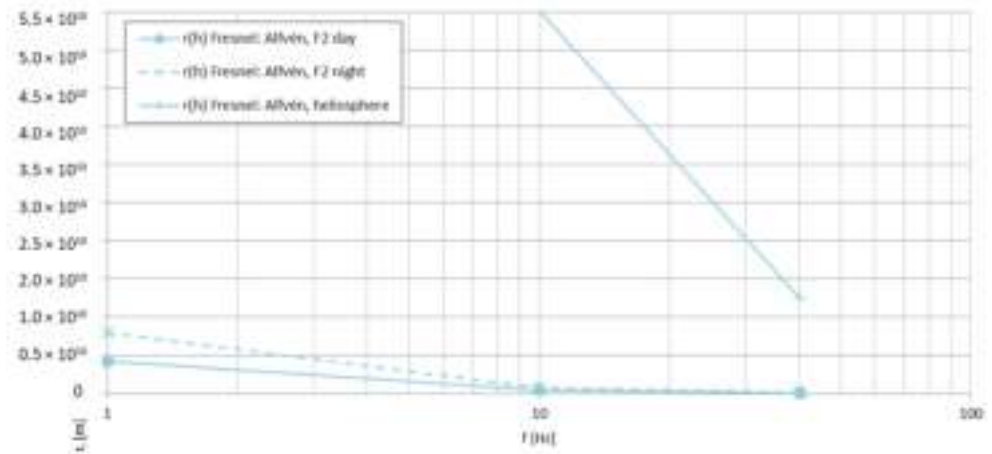


Figure 21. Calculated Fraunhofer radiation zones' radii for the Alfvén propagation mode (k_A) for $h = 1$ m and different electron concentrations.

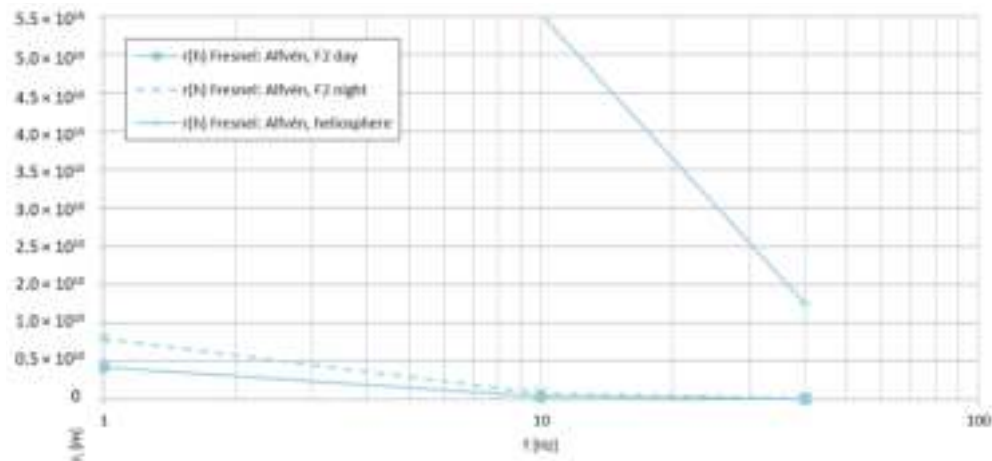


Figure 22. Calculated Fraunhofer radiation zones' radii for the Alfvén propagation mode (k_A) for $h = 10$ m and different electron concentrations.

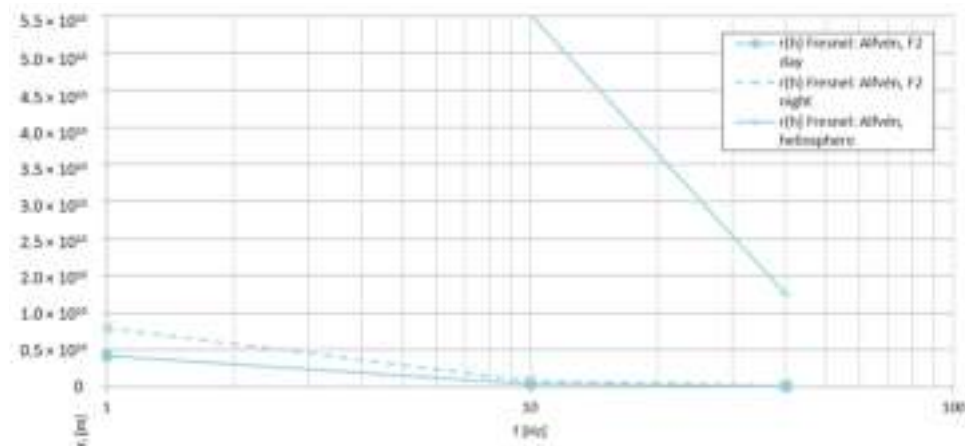


Figure 23. Calculated Fraunhofer radiation zones' radii for the Alfvén propagation mode (k_A) for $h = 100$ m and different electron concentrations.

It can be noticed that the approximate Fraunhofer zone radii (3) for heliospheric conditions remain many orders of magnitude higher for a given frequency than for the other electron-rich environments. Fraunhofer and Fresnel zones' radii calculated using precise formulas show mainly dependence on the antenna length and its order of magnitude while maintaining the general (and similar to the previous case) relation between heliospheric and electron-rich cases. The Fraunhofer zone case for the largest (100 m) antenna length starts to exhibit changes/radius length increases for the lower frequencies ($\sim$ kHz and lower), as the antenna length becomes more proportional to the signal's wavelength.

3. Evolution of the Antenna Radiation Pattern's Scalping

The scalping of the radiation (and, by the reciprocity theorem [40], reception) pattern of the antenna can be described as the length- or frequency-dependent appearance and relocation of minimums and maximums, redirecting the power transmitted by the antenna to certain directions or making the antenna more susceptible to signal reception at certain angles. If the exact solution [36] describing the number of maximums of the pattern—i.e., the lobes—is adapted to electron-rich conditions similarly to (3) and (7), the plasmatic solution for the number n_{Plasma} [-] of the scallops of the pattern can be defined as follows:

$$n_{Plasma} = \left\lceil h \sqrt{\frac{4}{\lambda^2} - \frac{e^2 n_e}{m_e \epsilon_0 \pi^2 c^2}} \right\rceil \quad (8)$$

where the parentheses define the upper limit of the formula (i.e., value above 1 gives $n_{Plasma} = 2$, value above 2 gives $n_{Plasma} = 3$ etc.).

Figures 24 and 25 present the number of the lobes (values of the formula within the parentheses) for the heliospheric environment ($n_e = 10^8 \text{ m}^{-3}$) for all considered propagation cases (classic approach, whistler and Alfvén) and all considered antenna lengths h (1, 10 and 100 m); Figures 26–28: for the ionospheric F-layer in daylight conditions ($n_e = 1.4 \times 10^{12} \text{ m}^{-3}$) and Figures 29–32: ionospheric F-layer for night conditions ($n_e = 4 \times 10^{11} \text{ m}^{-3}$).

It can be noticed that for the heliospheric conditions, the MHz-range of the signal's frequencies experiences an exponential rise in the number of pattern's lobes, which increases significantly with the elongation of the antenna; for other (lower) frequencies, the pattern's shape remains nearly constant. For the more electron-dense conditions (ionospheric F2 daytime), lower frequencies start to exhibit a strong exponential rise of the number of the pattern's lobes, apart from the Alfvén case, which remains nearly constant. A similar mechanism, but lower in achieved values, can be noticed for the ionospheric F2 nighttime electron concentration conditions; it can be linked to the antenna length becoming more

proportional to the signal's wavelength for the lower frequencies and becoming rapidly oversized for the signal's wavelength for the higher frequencies—the first mechanism can be noticed as 'amplified' by the increased electron density, and the latter mechanism appears as present globally.

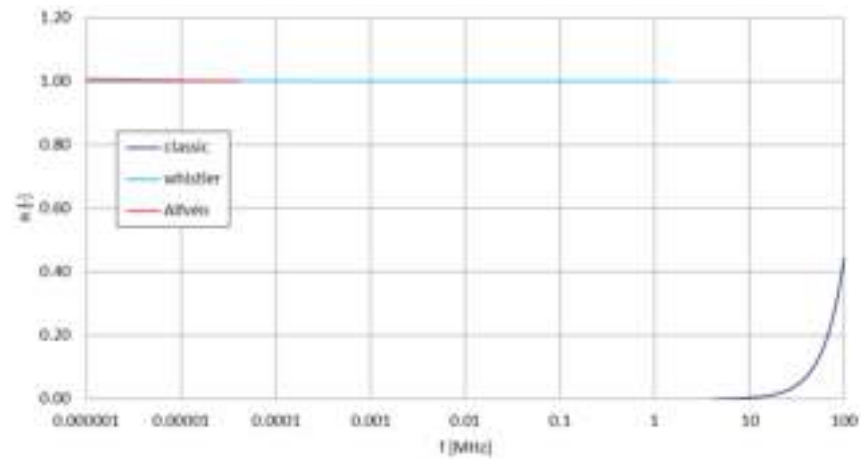


Figure 24. Calculated number of lobes of the antenna's radiation/reception pattern for antenna length $h = 1$ m and heliospheric environment ($n_e = 10^8 \text{ m}^{-3}$).

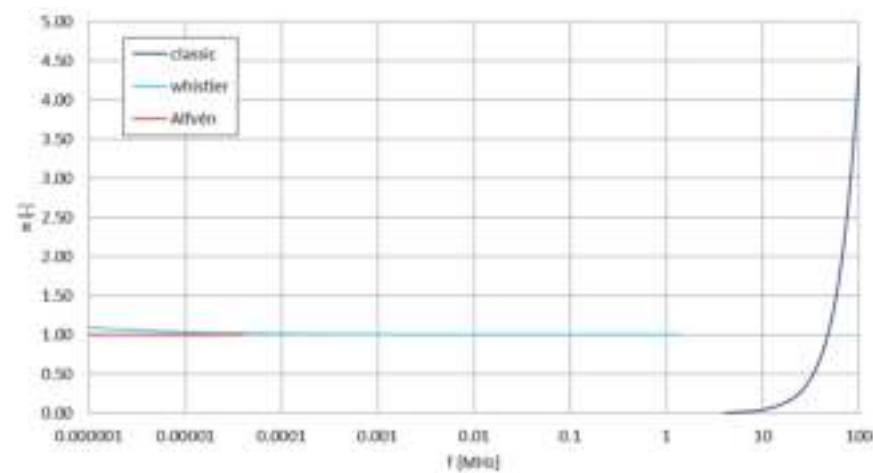


Figure 25. Calculated number of lobes of the antenna's radiation/reception pattern for antenna length $h = 10$ m and heliospheric environment ($n_e = 10^8 \text{ m}^{-3}$).

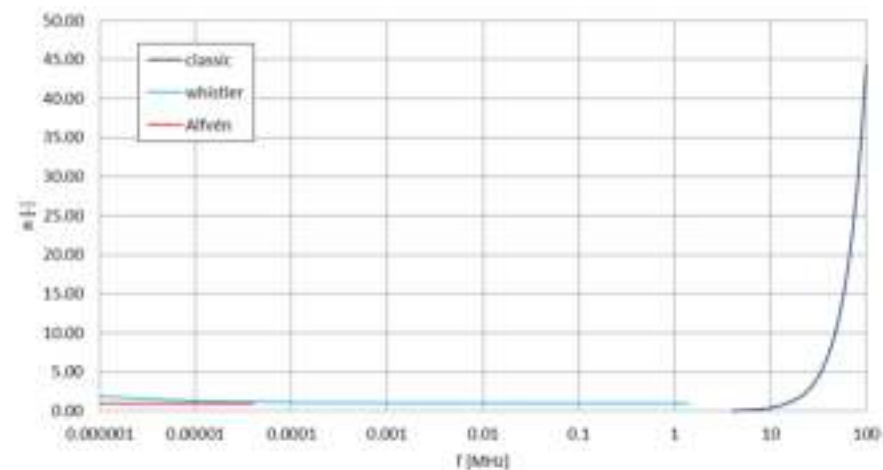


Figure 26. Calculated number of lobes of the antenna's radiation/reception pattern for antenna length $h = 100$ m and heliospheric environment ($n_e = 10^8 \text{ m}^{-3}$).

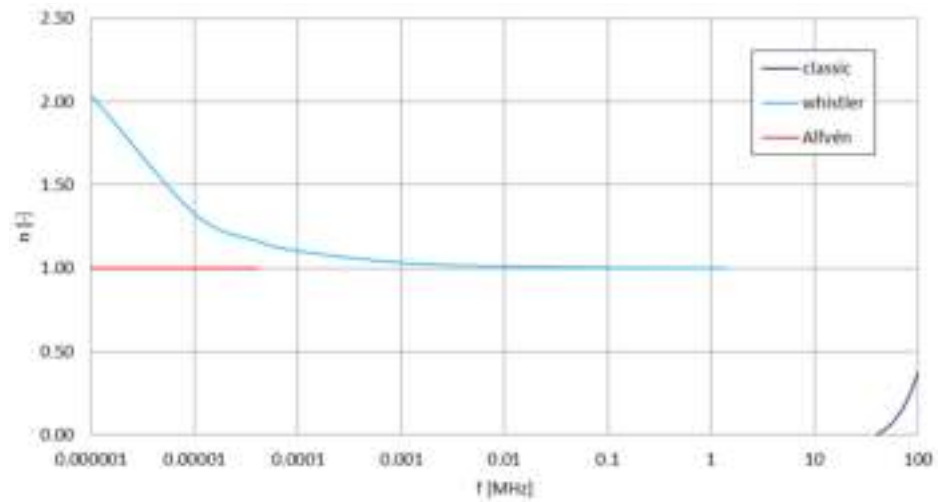


Figure 27. Calculated number of lobes of the antenna's radiation/reception pattern for antenna length $h = 1$ m and ionospheric F2-layer daylight environment ($n_e = 1.4 \times 10^{12} \text{ m}^{-3}$).

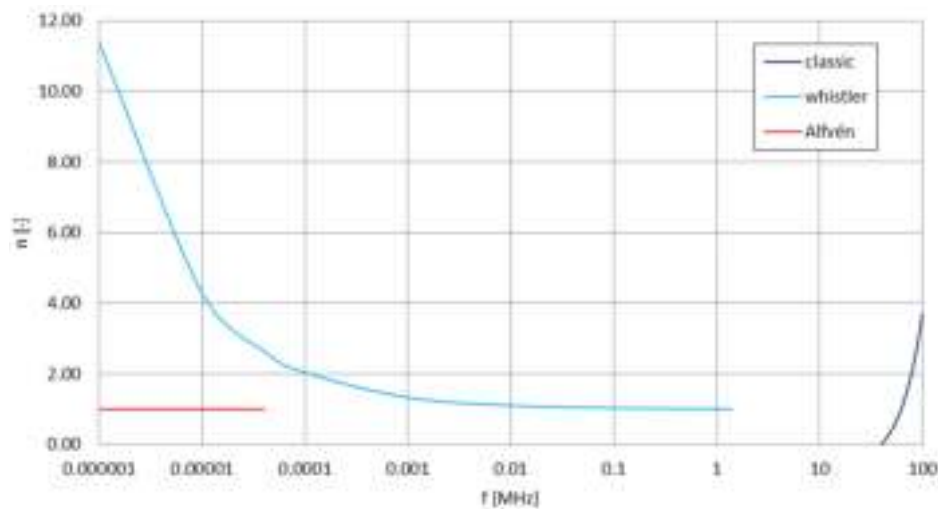


Figure 28. Calculated number of lobes of the antenna's radiation/reception pattern for antenna length $h = 10$ m and ionospheric F2-layer daylight environment ($n_e = 1.4 \times 10^{12} \text{ m}^{-3}$).

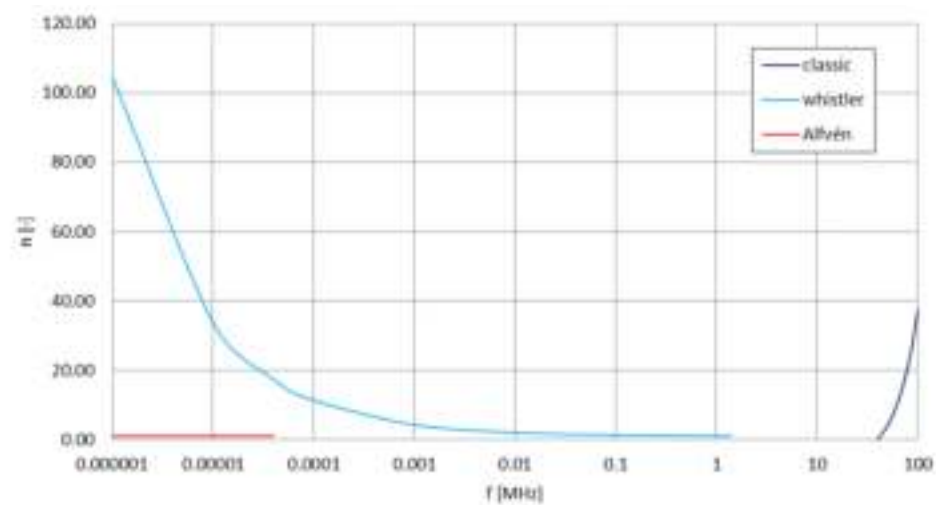


Figure 29. Calculated number of lobes of the antenna's radiation/reception pattern for antenna length $h = 100$ m and ionospheric F2-layer daylight environment ($n_e = 1.4 \times 10^{12} \text{ m}^{-3}$).

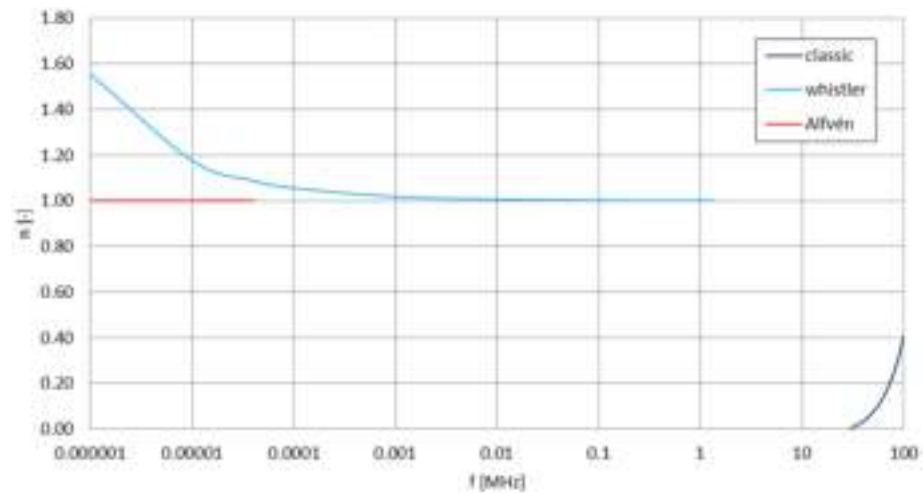


Figure 30. Calculated number of lobes of the antenna's radiation/reception pattern for antenna length $h = 1$ m and ionospheric F2-layer nighttime environment ($n_e = 4 \times 10^{11} \text{ m}^{-3}$).

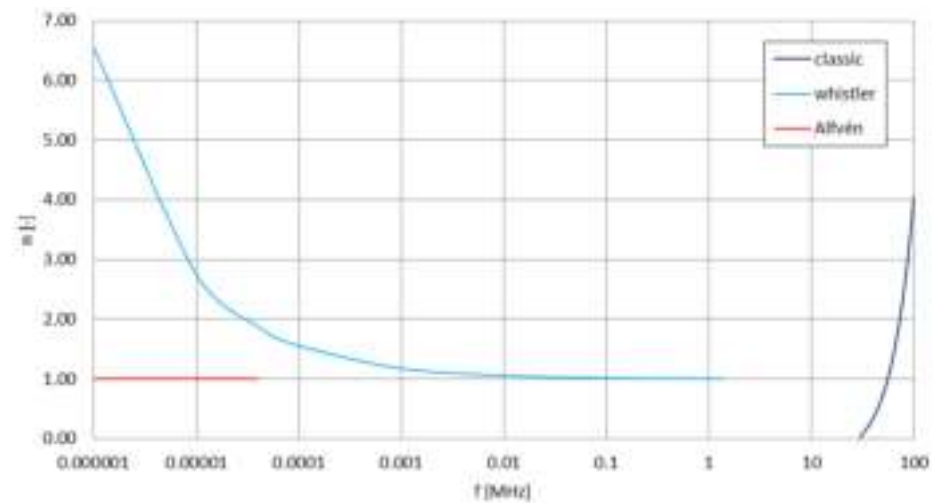


Figure 31. Calculated number of lobes of the antenna's radiation/reception pattern for antenna length $h = 10$ m and ionospheric F2-layer nighttime environment ($n_e = 4 \times 10^{11} \text{ m}^{-3}$).

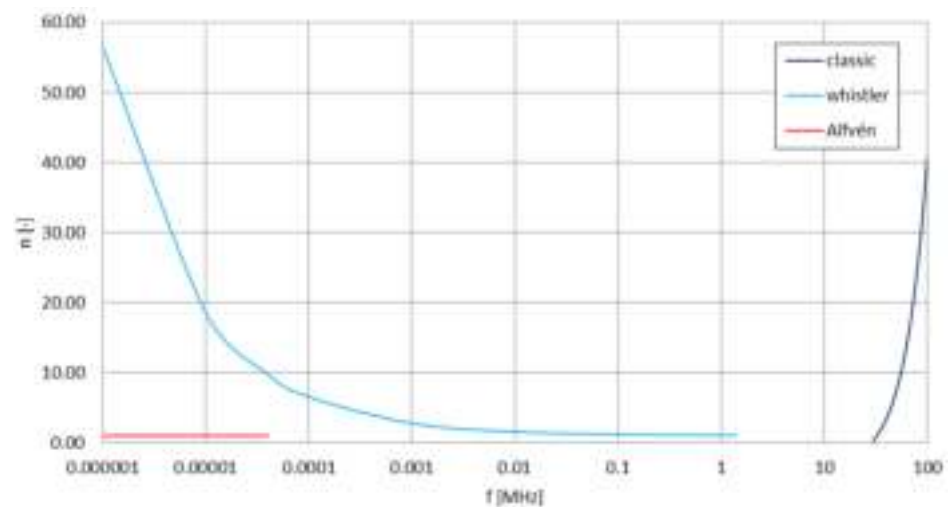


Figure 32. Calculated number of lobes of the antenna's radiation/reception pattern for antenna length $h = 100$ m and ionospheric F2-layer nighttime environment ($n_e = 4 \times 10^{11} \text{ m}^{-3}$).

4. Evolution of the Antenna Radiation Pattern's Directivity

The parameter of antenna's directivity D [-] can be explained as the ability of the antenna to form a concentrated beam of radiated energy at a certain angular width—defined as the intensity of radiation at a given direction/angle divided by the averaged radiation intensity over the entire radiation pattern (2π). Together with the antenna's efficiency η_A [-] (which depends on the radiation resistance of the antenna and its loss resistance), it gives the antenna's gain G [-]: $G = D\eta_A$ [40]; the antenna's gain is therefore directly proportional to the directivity, with the antenna's efficiency heavily dependent on the radiation losses in the actual spacecraft application (the radiation resistance for a dipole as a function of length-to-wavelength ratio remains well known [40]).

The directivity formula can be expressed as [36] follows:

$$D_i = 2 \left(\frac{1}{3} - \frac{\cos(2hk_i)}{(2hk_i)^2} + \frac{\sin(2hk_i)}{(2hk_i)^3} \right)^{-1} \quad (9)$$

with k_i defined for each case (ionospheric, whistler or Alfvén) using Formulas (5), (6) and (7). Figures 33–35 present the directivity calculated for the classic (7) case for different previously considered electron concentrations and antenna lengths $h = 1, 10$ and 100 m; similarly, Figures 36–38 employ the whistler wavenumber (5) and Figures 39–41: the Alfvén wavenumber (6).

For the classic-ionospheric cases, the longer the antenna, the more constant the directivity, reaching a value of 6 for high-MHz frequencies; for the kHz-and-lower frequency range, the directivity reaches values below 1, which indicates a lack of a well-formed separate lobes in the radiation pattern (as in the Figures in the previous paragraph). For the whistler propagation mode, the general behavior of the directivity changes moves upward with the frequency values with the increase in the antenna length; both ionospheric conditions show two maximums at about 6.5, while the heliospheric case starts to give larger values at the largest considered antenna length of 100 m. For the Alfvén propagation mode and frequency range, in Figure 40, the heliospheric conditions produced numerical errors in Formula (9) and were not included in the final plot; the smaller the antenna, the more susceptible it appears to higher electron concentrations, with the largest considered antenna of 100 m showing high variability to the lowest (heliospheric) electron concentration. For all the Alfvén cases, the maximum directivity value reached 4.1.

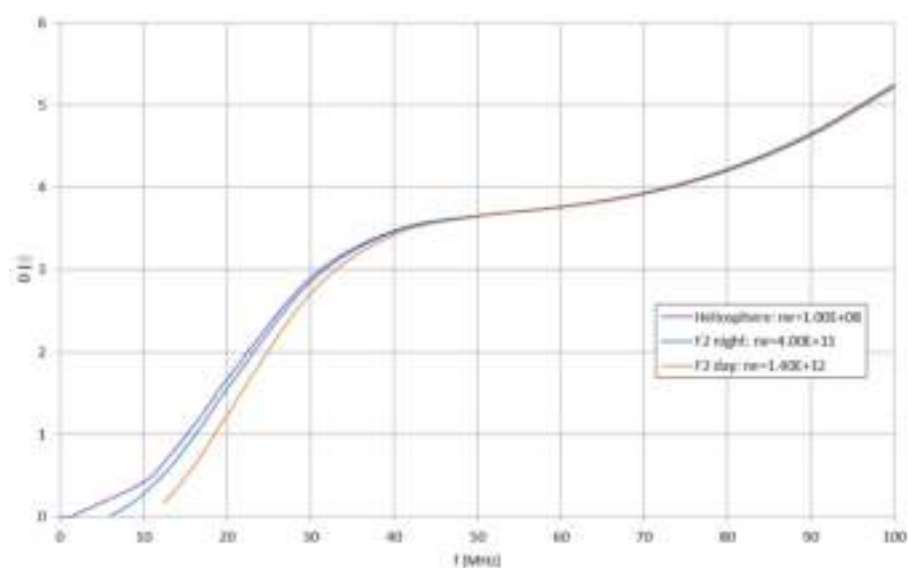


Figure 33. Calculated antenna directivity for antenna length $h = 1$ m and wavenumber k_f (7).

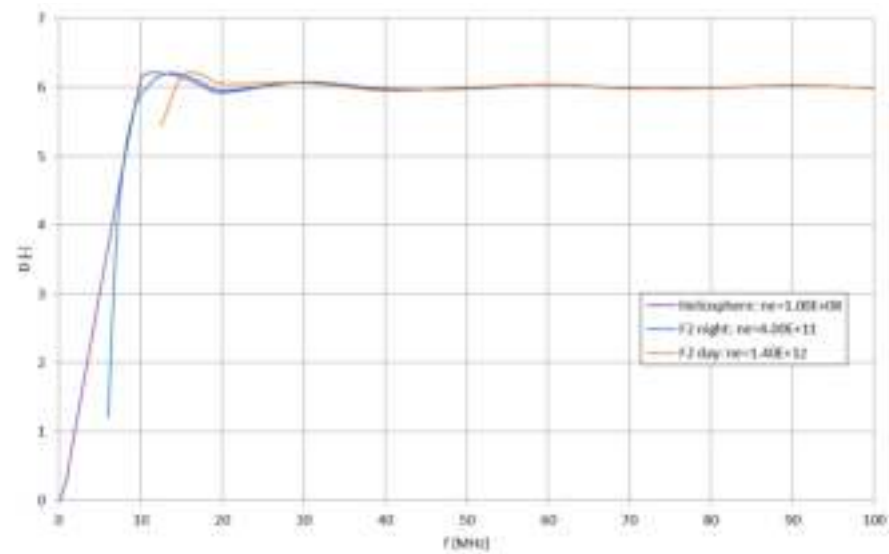


Figure 34. Calculated antenna directivity for antenna length $h = 10$ m and wavenumber k_I (7).

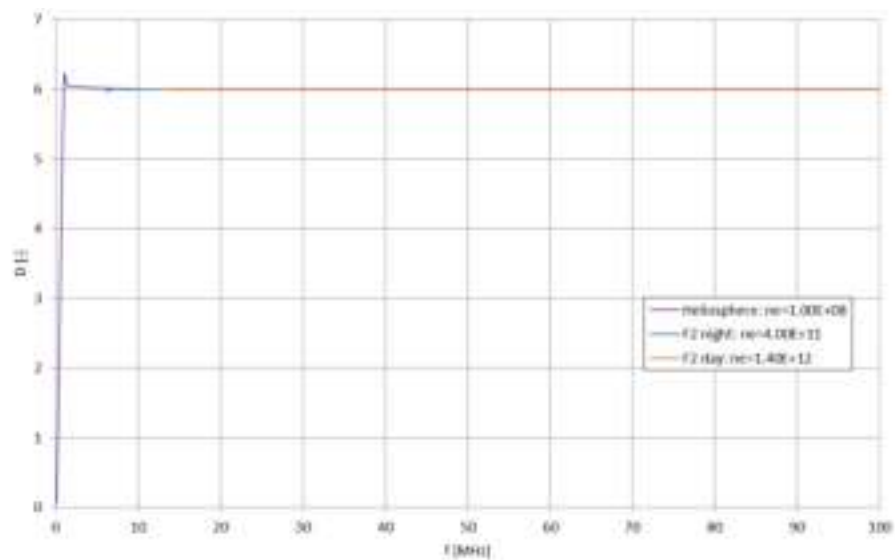


Figure 35. Calculated antenna directivity for antenna length $h = 100$ m and wavenumber k_I (7).

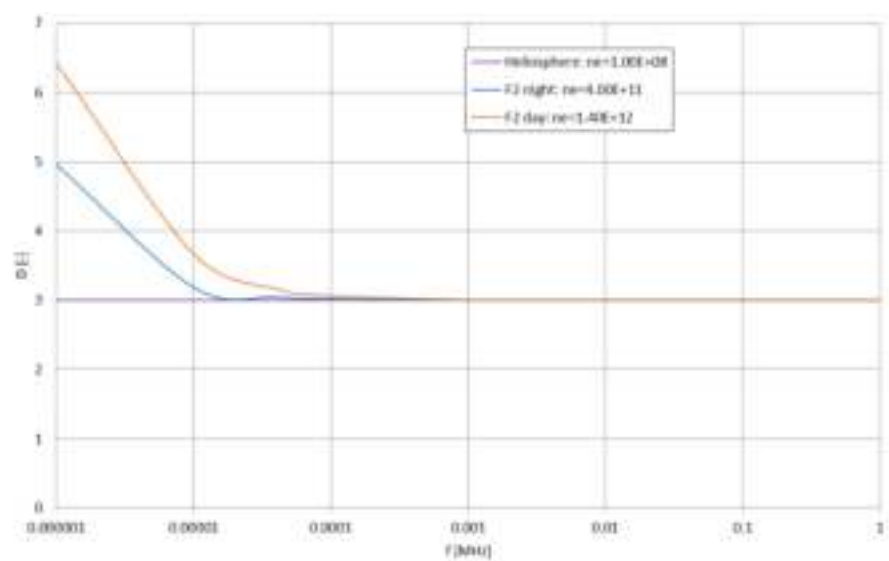


Figure 36. Calculated antenna directivity for antenna length $h = 1$ m and wavenumber k_W (5).

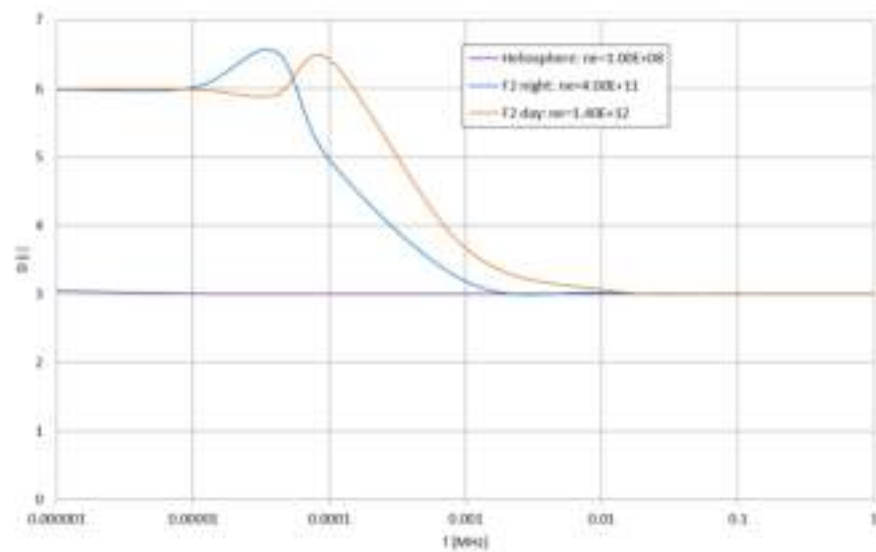


Figure 37. Calculated antenna directivity for antenna length $h = 10$ m and wavenumber k_W (5).

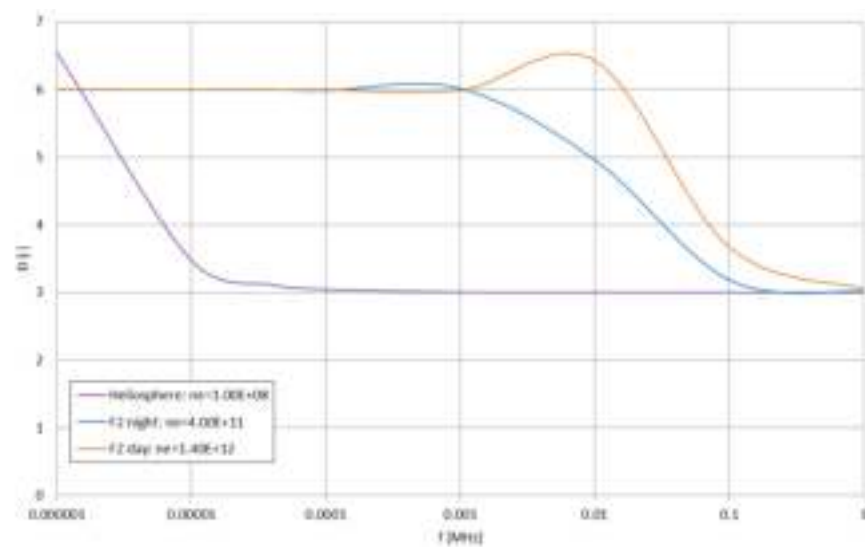


Figure 38. Calculated antenna directivity for antenna length $h = 100$ m and wavenumber k_W (5).

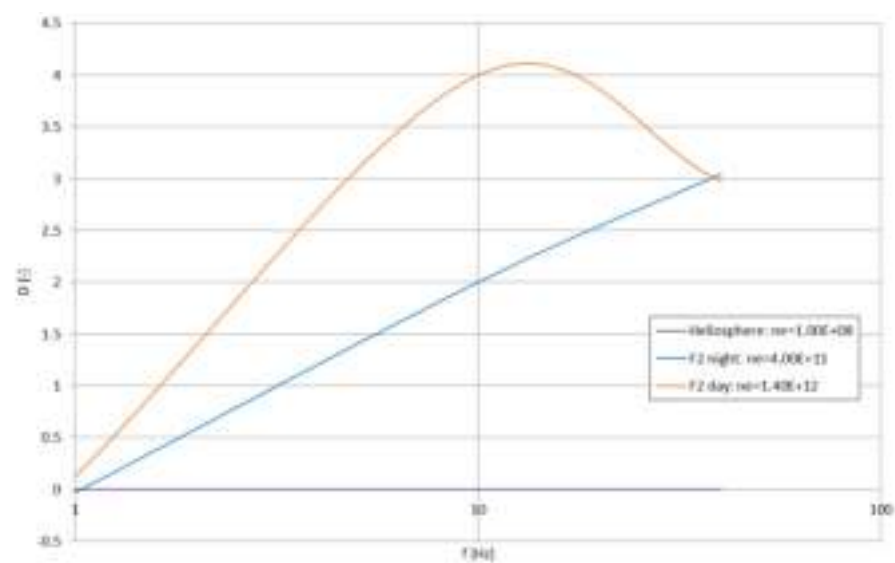


Figure 39. Calculated antenna directivity for antenna length $h = 1$ m and wavenumber k_A (6).

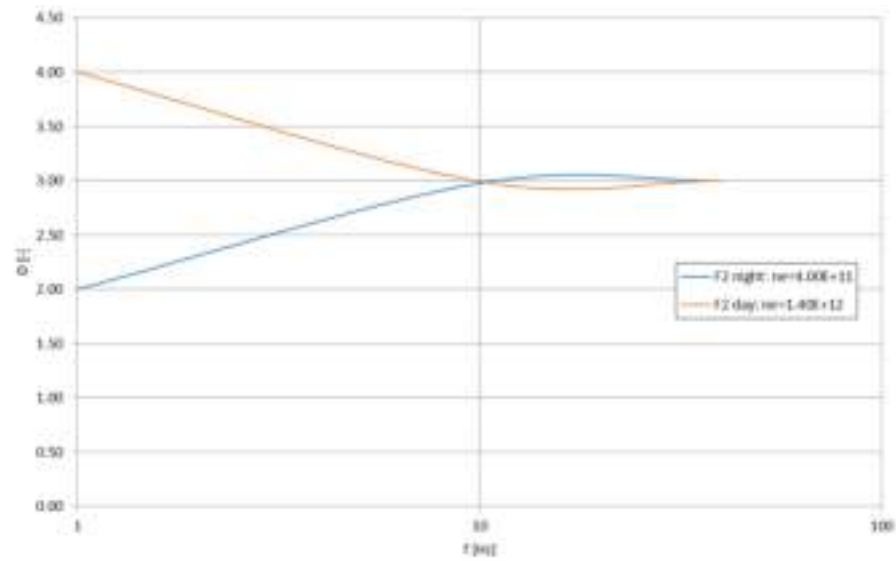


Figure 40. Calculated antenna directivity for antenna length $h = 10$ m and wavenumber k_A (6).

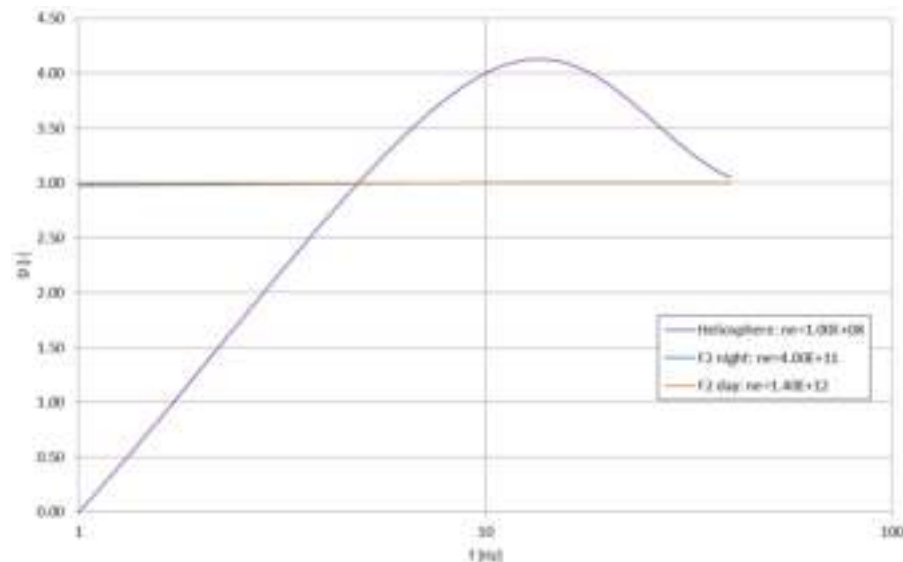


Figure 41. Calculated antenna directivity for antenna length $h = 100$ m and wavenumber k_A (6).

5. Radiation Pattern Transformation Coefficient from the Air/Vacuum to Plasmatic Conditions

The experimental verification of the shape of the antenna radiation/reception pattern, which always incorporates the actual mechanical configuration of the antenna and the antenna mounting (e.g., on a spacecraft), is mainly possible either for the terrestrial atmospheric conditions or close-to-vacuum conditions (if additional heat transfer, typical for space environment, is required to be re-created for other components). Analytical solutions for the radiation/reception pattern of a given antenna submerged in the plasma of given parameters have been developed—e.g., the radiation pattern by Bannister et al. for the i -th wavenumber k [35]:

$$f_P(\theta) = \frac{\sin \theta}{1 - \cos \theta} \sin \left[\frac{1}{2} k_i h (1 - \cos \theta) \right] \quad (10)$$

where θ [rad] is the angle between the symmetry plane of the antenna (considered as a symmetrical dipole of the total length h [m]) and the main axis of the antenna. This radiation pattern shows the common (for large plasmatic antennas) property of radiating the energy over a direction close to the main axis of the antenna, which manifests itself stronger with the increasing length of the antenna [24,34,35] and remains totally inconsistent with the shape of the pattern/antenna behavior in non-plasmatic conditions [23]. This

property remains of interest not solely for the antenna testing purposes and the actual in-plasma transmitting antenna applications, but also for the receiving antennas submerged in plasmas, which—as their reception pattern evolves in relation to the plasma around them—may change in a way which shall exclude some of the directions to the antenna from receiving any signal; this shall remain useful for the accurate description of, e.g., the incoming directions and actual signal strengths of the Auroral Kilometric Radiation.

A method based on a simple transformation of the analytical definitions of the radiation pattern's shape from the air/vacuum conditions to the plasmatic conditions—with the assumption that this approach remains valid for the experimentally defined pattern shapes—for a given θ angle value can be defined as follows:

$$T_{\theta=const.}(f_{A\ real}(\theta) \rightarrow f_{P\ real}(\theta)) : f_{P\ real} = f_{A\ real}(\theta) \frac{f_P(\theta)}{f_A(\theta)} \quad (11)$$

where $f_i(\theta)$ are the functions of the antenna radiation patterns and the subscripts P , A and $real$ describe air/vacuum, plasmatic and real conditions, subsequently. The ratio of the plasmatic-to-air/vacuum radiation pattern functions requires an additional division of each $f_i(\theta)$ function by the maximum value reached by it, in order to make the subsequent cases comparable in values (this would, however, erase the accurate information on the directivity). Naturally, for this transformation, the value of $f_A(\theta)$ must not reach zero (may reach a real value of $\varepsilon \rightarrow 0$) and remains valid for the condition below (otherwise it is a tautology):

$$\int_{\varepsilon \rightarrow 0}^{\frac{\pi}{2}-\varepsilon} f_A(\theta) d\theta \neq \int_{\varepsilon \rightarrow 0}^{\frac{\pi}{2}-\varepsilon} f_P(\theta) d\theta \quad (12)$$

For the actual space-applied conditions and measured different radio signals mentioned in paragraph 1, the precise direction of the incoming signal is not always known, and the evolution of the antenna radiation/reception pattern for different plasmatic conditions shall introduce even more variables to its definition. If a two-dimensional frequency spectrum is obtained from such antenna, and basic plasma parameters are accessible (as described in paragraph 2), the ratio from the Formula (11) can be expanded to form a general frequency- and angle-dependent coefficient $w(\theta, f)$, which includes the plasmatic radiation pattern (10) and air/vacuum radiation pattern for short linear antenna [36]; the differences between the air/vacuum and plasmatic antenna radiation patterns are defined in the space around the antenna wire and over the entire possible range of θ (with the integration-facilitating assumption that the antenna is a symmetrical dipole of the total length h):

$$w(\theta, f) = 4\pi \int_{\varepsilon \rightarrow 0}^{\frac{\pi}{2}-\varepsilon} \frac{f_P(\theta)}{\max[f_P(\theta)]} \left(\frac{f_A(\theta)}{\max[f_A(\theta)]} \right)^{-1} d\theta = 4\pi \int_{\varepsilon \rightarrow 0}^{\frac{\pi}{2}-\varepsilon} \frac{\sin\left[\frac{1}{2}k_i h(1 - \cos\theta)\right]}{\sin\theta(1 - \cos\theta) \cos^2\left(\frac{2\pi f h}{c} \cos\theta\right)} \frac{\max[f_A(\theta)]}{\max[f_P(\theta)]} d\theta \quad (13)$$

An approximate solution can be computed as follows:

$$w(\theta, f) \approx \sum_{i=1}^n \frac{1}{2} \theta_i \frac{\max[f_A(\theta)]}{\max[f_P(\theta)]} \left(\frac{\sin\left[\frac{1}{2}k_i h(1 - \cos\theta_i)\right]}{\sin\theta_i(1 - \cos\theta_i) \cos^2\left(\frac{2\pi f h}{c} \cos\theta_i\right)} + \frac{\sin\left[\frac{1}{2}k_i h(1 - \cos\theta_{i-1})\right]}{\sin\theta_{i-1}(1 - \cos\theta_{i-1}) \cos^2\left(\frac{2\pi f h}{c} \cos\theta_{i-1}\right)} \right) \quad (14)$$

As the wavenumbers k_i may incorporate the B_0 magnetic field compound, it is necessary to approximate this compound in a way which reflects the structure and direction of the magnetic field around the antenna. For a case where the magnetic field induction remains constant along the Z axis of the antenna, the B_0 parameter can be formed as follows:

$$B_0 = \int_0^{2\pi} B_{0\ max.} |\sin\phi| d\phi = 4B_{0\ max.} \quad (15)$$

where ϕ [rad] is the angle laying in the plane parallel to the plane of symmetry of the dipole. If B_0 remains not constant along the antenna wire and has a known distribution function with the maximum value of $B_{0\max}$, and the argument of z , the formula is presented as follows:

$$B_0(z) = \int_0^{\frac{h}{2}} \int_0^{2\pi} B_{0\max}(z) |\sin \phi| d\phi dz = 4 \int_0^{\frac{h}{2}} B_{0\max}(z) dz \quad (16)$$

Figures 42–45 present the $w(f)$ coefficient (after the integration over θ) calculated for a wide frequency range for different electron concentrations (as in the previous paragraphs) and subsequent antenna lengths: 1, 10 and 100 m. As the Formulas (13) and (14) incorporate the Formula (10), which is essentially valid for the whistler propagation mode frequency range, the maximum frequency in these figures does not exceed the value of 1.4 MHz—the calculation beyond this limit would be possible after the clarification of the accuracy of Formula (10) above this frequency.

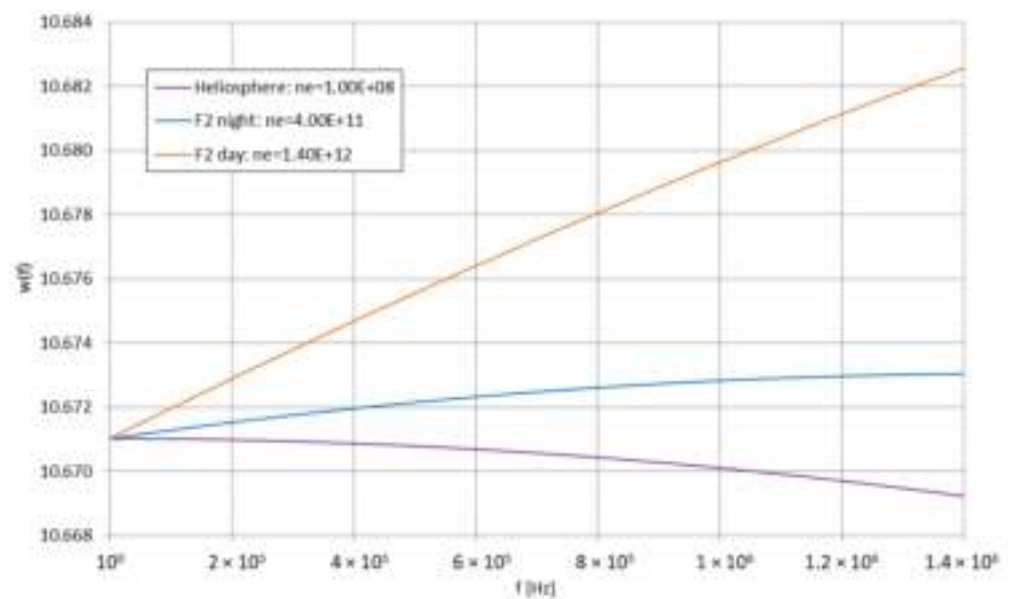


Figure 42. The calculated $w(f)$ transformation parameter for antenna length $h = 1$ m.

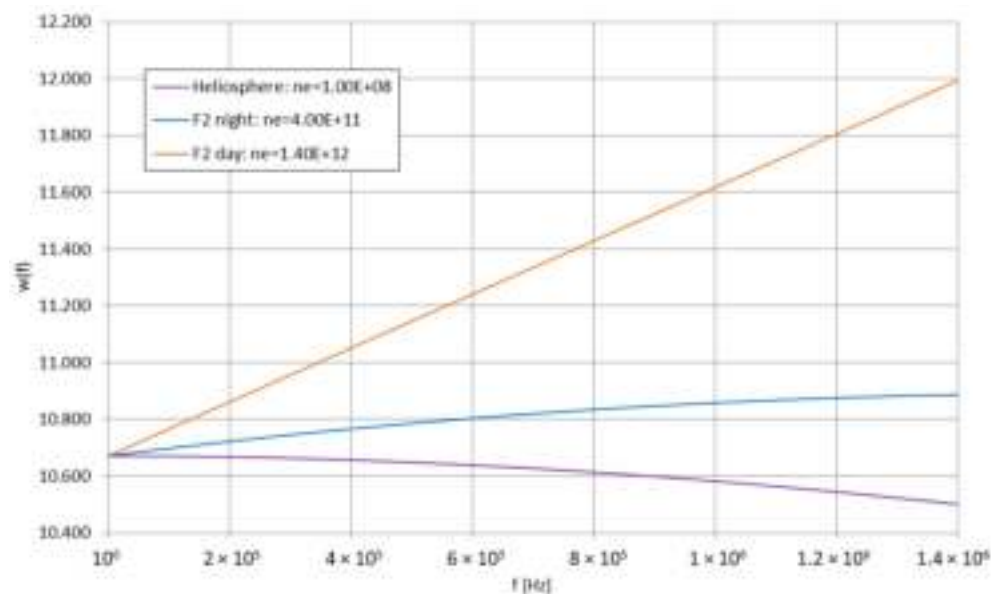


Figure 43. The calculated $w(f)$ transformation parameter for antenna length $h = 10$ m.

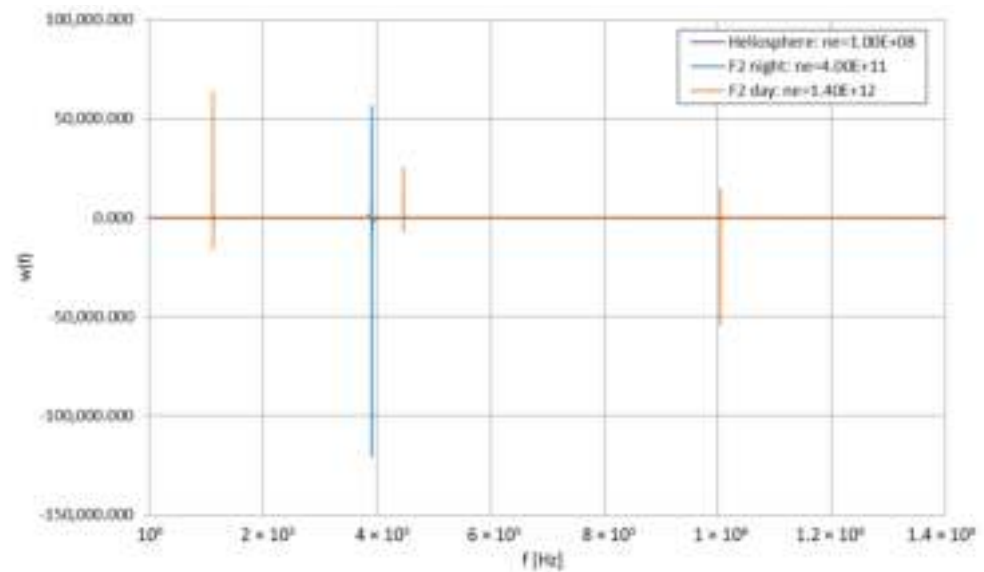


Figure 44. The calculated $w(f)$ transformation parameter for antenna length $h = 100$ m.

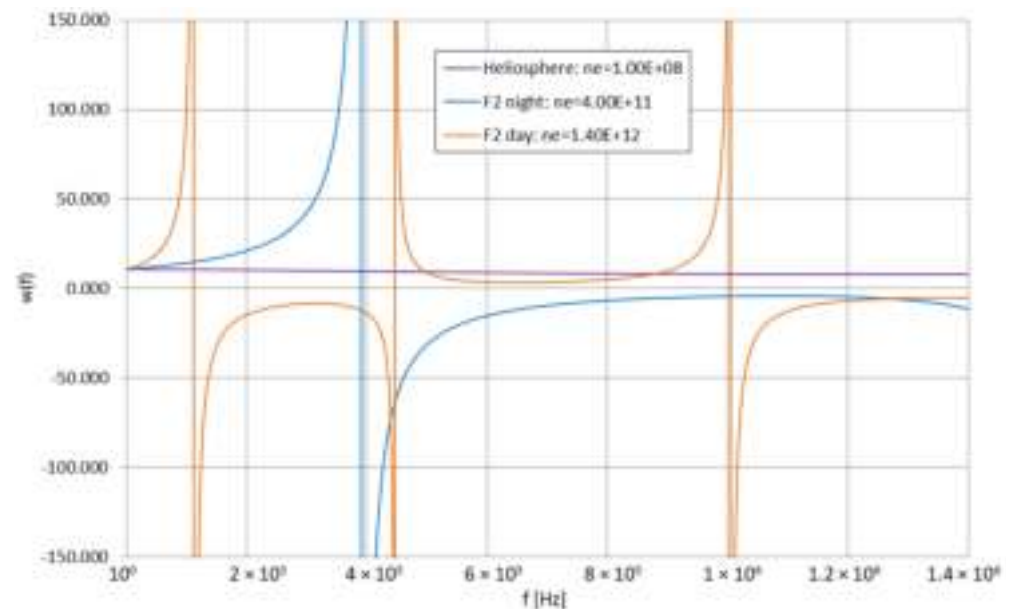


Figure 45. The calculated $w(f)$ transformation parameter for antenna length $h = 100$ m—enlarged.

For the smaller antenna lengths, the calculated coefficients demonstrate a nearly constant behavior, slightly rising for the higher frequencies for denser electron conditions (F2 layer in daylight) and slightly decreasing for the less dense electron conditions (heliospheric). For the largest considered antenna length of 100 m, the coefficient shows multiple significant changes in values for the more electron-dense conditions, which demonstrate the differences between the antenna radiation patterns for air/vacuum and plasmatic cases. As the coefficient is meant to bring the air/vacuum case in value to the plasmatic case, it reacts on the appearance of the lobes in the pattern, including any sign switching of the function, which—from a mathematical point of view—may appear.

6. Discussion

The presented evolutions of the basic antenna parameters describing its abilities to radiate or receive electromagnetic energy clearly shows the dependence on the electrical conditions or the surrounding space—as presented, these conditions do not only influence the impedance of the antenna (as shown in various previous research from the past decades),

but its functioning in space as energy radiators or energy receivers. It is important from two major aspects—the aspect of the spacecraft and antenna design and the aspect of interpreting and using data from the existing spacecraft and its instrumentation.

The design aspect concentrates mainly on the allocation of the antennas in the proximity of the spacecraft in a way which would prevent the measurements of the plasma influenced directly by the spacecraft (appearing around the structure of the spacecraft—unless this is a specific requirement for the mission). This direct influence can be linked to the presence of the conductive structures (booms/beams, solar panels, spacecraft frames etc.) within the Fresnel or the Fraunhofer zones around the antenna.

The zones' radiuses, calculated using different formulas in paragraph 2, clearly show the evolution with the changing plasma parameters—they remain highly sensitive to different electron concentrations n_e . For the smaller antennas ($\sim 10^0$ m in length), all the zones' radiuses are relatively condensed; it is therefore not difficult to include in the instrumentation mechanical design requirements the requirement of placing larger structures beyond the Fraunhofer zone radius (up to a maximum frequency of interest). With the increasing length of the antenna ($\sim 10^1$ – 10^2 m), the accurate (formula-based) Fraunhofer zone becomes too large to be easily included as a dimensional requirement; therefore, the Fresnel zone radiuses—or the approximate formula for the Fraunhofer zone radius, derived from the characteristics of the EM fields around the antenna—can be considered.

For a case where the spacecraft is already designed, built and in operation, these radiuses can be used as indicators whether it would be beneficial to verify the actual radiation/reception pattern of the antenna in a way which includes the spacecraft mechanical elements as passive antenna reflectors. This is especially important for the reception of the radio signals propagating in the whistler and Alfvén-wave modes, which have significantly larger Fraunhofer and Fresnel zones' radiuses, ranging to the size of an entire satellite constellation.

The rise of the electron concentration n_e also influences the scalloping of the antenna radiation/reception pattern; for the highest considered concentrations, proportional to those in the ionospheric F2 layer in daylight conditions (photoionization etc.), the appearance of the separate pattern lobes (maximums and minimums) is present for both the whistler propagation mode at lower frequencies ($\sim$ kHz and lower) and for the high MHz frequencies. The increase in scalloping is also directly related to the proportion of the incoming signal's wavelength to the antenna's length, which manifests itself simultaneously with the influence of the growing n_e .

Sufficiently prominent lobes shall elevate the parameter of directivity, which can be used—together with the antenna's efficiency parameter—to define the antenna gain, or simply to describe the effectiveness of the antenna radiation of reception at a given frequency in a plasma of given parameters. The behaviors of the directivity are significantly different for every propagation mode case and every considered n_e value—this requires a careful approach for a precise set of measurement requirements, e.g., frequency range and expected plasma parameters. Generally, it can be noticed that smaller antennas (length proportional to 10^0 m) are more prone to directivity changes, especially for high-MHz frequency ranges—similarly to the scalloping behavior noticed for these antennas. For high values of n_e , the whistler propagation mode exhibits two global directivity maximums, appearing in every considered antenna length case, which are nonexistent in the low-electron-content environment.

Figure 45 clearly shows the changes the plasmatic environment produces in antenna radiation/reception patterns for large antenna apertures ($\sim 10^2$ m)—the presented coefficient aimed to transform the air/vacuum pattern into the plasmatic pattern reflects the evolved plasmatic pattern, rich in maximums and minimums (actual plasmatic radiation pattern plots, showing the apparent radiation along the antenna's main axis, can be found, e.g., in [35]). It proves that the measurements on the ground do not give a full understanding of the antenna operation in the space environment with significant electron concentration.

The proposed method of pattern transformation, however, allows a basic description of the approximated antenna behavior in plasma, which shall be useful for the analysis of the spectral electromagnetic data from existing satellites—the definition of the, e.g., direction of the received signal’s propagation or its actual strength (bearing in mind the varying antenna directivity and scalloping).

7. Conclusions

In this paper, the basic antenna parameters related to its ability to radiate or receive electromagnetic energy are presented as variables depending on the simplified conditions of the plasma in which the antenna is immersed. Three propagation modes are considered (classic-ionospheric, whistler and Alfvén), along with three exemplary electron concentrations (generalized heliospheric and ionospheric F2 layer: daylight and nighttime) and a constant magnetic field induction. The antenna parameters considered are the radiation zones (inductive, Fresnel, Fraunhofer), scalloping and directivity.

It is shown that these parameters evolve intensely with the changing plasmatic conditions. The radiation zones remain condensed for the smaller sizes of the antennas and expand significantly for the larger antenna lengths, which can be treated as a technical disadvantage despite the increase in the antenna’s aperture. Also, for larger antennas, the Fresnel zones become the largest physically applicable zones. The scalloping (number of lobes in the radiation pattern) is increased both by the increasing the electron density and the antenna length-to-wavelength ratio.

This paper also proposes a method of transformation of the antenna radiation/reception pattern as measured on the ground to the one which would be expected in the space plasma environment. This method is shown as effectively operating for larger antenna lengths (>10 m), providing a useful transformation of antenna radiation pattern from the air/vacuum environment (testing environment) to expected plasmatic conditions.

The presented data—in a convenient form of plots for different parameters and variables, or in a form of formulas easy to adapt to individual plasma cases—effectively complement the existing work on plasmatic antenna’s impedances, currents and sheaths and can be used for the actual assessment of the antenna performance in existing space weather satellites. This shall allow the enhanced triangulation of the registered incoming signals, as well as additional requirements/constraints for the new satellite’s instrumentation design, which shall be helpful in, e.g., moving the measuring equipment away from the direct influence of the plasma appearing around the spacecraft itself.

Funding: This research received no external funding.

Data Availability Statement: Data are available upon request.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Green, J.L.; Inan, U.S. Lightning Effects on Space Plasmas and Applications. In *Plasma Physics Applied*; Research Signpost; Grabbe, C., Ed.; American Physical Society: College Park, MD, USA, 2007.
2. Morozova, E.A.; Mogilevsky, M.M.; Hanasz, J.; Rosanov, A.A. The fine structure of the AKR electromagnetic field as measured by the Interball-2 satellite. *Cosm. Res.* **2002**, *40*, 404–410. [\[CrossRef\]](#)
3. Chuchra-Konrad, A.; Matyjasiak, B.; Schreiber, R.; Rothkaehl, H. A correlation between AKR-like emissions and field-aligned currents. In Proceedings of the URSI GASS 2023, Sapporo, Japan, 19–26 August 2023.
4. Bresler, K.; Zawila, R.; Ciarka, M.; Shmyk, A.; Miś, T.A.; Mystkowski, R.; Martyniak, J.; Wiater, K. TOTORO Test Observations of Transient Objects and RadiO on board BEXUS 33. In Proceedings of the 26th ESA PAC Symposium on European Rocket and Balloon Programmes and Related Research, Lucerne, Switzerland, 19–23 May 2024.
5. Gurnett, D.A.; Anderson, R.R.; Scarf, F.L.; Fredricks, R.W.; Smith, E.J. Initial results from the ISEE-1 and -2 plasma wave investigation. *Space Sci. Rev.* **1979**, *23*, 103–122. [\[CrossRef\]](#)
6. Gallet, R.M.; Helliwell, R.A. Origin of “Very-Low-Frequency Emissions”. *J. Natl. Bur. Stand.—D Radio Propag.* **1959**, *63D*, 21–27. [\[CrossRef\]](#)
7. Ferencz, C.; Lichtenberger, J.; Hamar, D.; Ferencz, O.E.; Steinbach, P.; Székely, B.; Parrot, M.; Lefeuvre, F.; Berthelier, J.-J.; Clilverd, M. An unusual VLF signature recorded by the DEMETER satellite. *J. Geophys. Res.* **2010**, *115*, A02210. [\[CrossRef\]](#)

8. Singh, D.; Singh, A.K.; Patel, R.P.; Singh, R.; Singh, R.P.; Veenadhari, B.; Mukherjee, M. Thunderstorms, Lightning, Sprites and Magnetospheric Whistler-Mode Radio Waves. *Surv. Geophys.* **2008**, *29*, 499–551. [\[CrossRef\]](#)
9. Xiao, F.; Zhou, Q.; Su, Z.; He, Z.; Yang, Z.; Liu, S.; He, Y.; Gao, Z. Explaining occurrences of auroral kilometric radiation in Van Allen radiation belts. *Geophys. Res. Lett.* **2016**, *43*, 971–978. [\[CrossRef\]](#)
10. Kishore, A.; Deo, A.; Kumar, S. Upper Atmospheric Remote Sensing Using ELF-VLF Lightning Generated Tweek and Whistler Sferics. *South Pac. J. Nat. Appl. Sci.* **2016**, *34*, 12–20. [\[CrossRef\]](#)
11. Błęcki, J. ULF/ELF plasma turbulence associated with small scale structures in the vicinity of the Earth magnetopause and bow shock. In Proceedings of the 2nd National Symposium PLASMA '95 "Research and Applications of Plasmas", Warsaw, Poland, 26–28 June 1995; Invited Papers; Volume 2.
12. Mansilla, G.A. Ionospheric effects of an intense geomagnetic storm. *Stud. Geophys. Geod.* **2007**, *51*, 563–574. [\[CrossRef\]](#)
13. Šulić, D.M.; Srećković, V.A.; Mihajlov, A.A. A study of VLF signal variations associated with the changes of ionization level in the D-region in consequence of solar conditions. *Adv. Space Res.* **2016**, *57*, 1029–1043. [\[CrossRef\]](#)
14. Thomson, N.R.; Clilverd, M.A.; McRae, W.M. Nighttime Ionospheric D-region Parameters from VLF Phase and Amplitude. *J. Geophys. Res. Atmos.* **2007**, *112*. [\[CrossRef\]](#)
15. Žigman, V.; Dominique, M.; Grubor, D.; Rodger, C.J.; Clilverd, M.A. Lower-ionosphere electron density and effective recombination coefficients from multi-instrument space observations and ground VLF measurements during solar flares. *J. Atmos. Sol.-Terr. Phys.* **2023**, *247*, 106074. [\[CrossRef\]](#)
16. Jarvinen, R.; Kallio, E. Global hybrid modelling of Mercury's space weather and BepiColombo Observations. In Proceedings of the 4th URSI AT-RASC, Gran Canaria, Spain, 19–24 May 2024.
17. Mikhailov, Y.M.; Mikhailova, G.A.; Kapustina, O.V. ELF and VLF electromagnetic background in outer ionosphere over seismoactive regions (satellite "INTERCOSMOS-24"). In Proceedings of the International Symposium PLASMA '97 "Research and Applications of Plasmas", Jarnottówek Near Opole, Poland, 10–12 June 1997; Contributed Papers; Volume 1.
18. Klimov, S.; Romanov, S.; Savin, S.; Nikolaeva, N.; Skalsky, A.; Petrukovich, A.; Nozdrachev, M.; Triska, P.; Amata, E.; Ciobanu, M.; et al. INTERBALL tail probe. Magnetospheric plasma boundaries study. In Proceedings of the International Symposium PLASMA '97 "Research and Applications of Plasmas", Jarnottówek Near Opole, Poland, 10–12 June 1997; Contributed Papers; Volume 1.
19. Błęcki, J.; Kosacki, K.; Wronowski, R.; Savin, S.; Simunek, J.; Smilauer, J.; Juchniewicz, J.; Klimov, S.; Triska, P. Electrostatic and electromagnetic components of the plasma wave observed in the polar cusp onboard Prognost-8 and Magion-4 satellites. In Proceedings of the International Symposium PLASMA '97 "Research and Applications of Plasmas", Jarnottówek Near Opole, Poland, 10–12 June 1997; Contributed Papers; Volume 1.
20. Błęcki, J.; Kosacki, K.; Wronowski, R.; Nemecek, Z.; Safrankova, J.; Simunek, J.; Smilauer, J.; Borodkova, N.; Romanov, S.A.; Savin, S.P.; et al. Electron-cyclotron harmonics observed by Magion-4 subsatellite in the plasma sheet. In Proceedings of the International Symposium PLASMA '97 "Research and Applications of Plasmas", Jarnottówek Near Opole, Poland, 10–12 June 1997; Contributed Papers; Volume 1.
21. Rothkaehl, H.; Błęcki, J.; Klos, Z. The signature of ionospheric-magnetospheric coupling investigated with the plasma wave package PWP on the board of CESAR satellite. In Proceedings of the International Symposium PLASMA '97 "Research and Applications of Plasmas", Jarnottówek Near Opole, Poland, 10–12 June 1997; Contributed Papers; Volume 1.
22. Wooliscroft, L.J.C. Plasma wave measurements with the ESA/NASA CLUSTER mission. In Proceedings of the 2nd National Symposium PLASMA '95 "Research and Applications of Plasmas", Warsaw, Poland, 26–28 June 1995; Invited Papers; Volume 2.
23. McCollough, J.P.; Miyoshi, Y.; Ginet, G.P.; Johnston, W.R.; Su, Y.-J.; Starks, M.J.; Kasahara, Y.; Kojima, H.; Matsuda, S.; Shinohara, I.; et al. Space-to-space very low frequency radio transmission in the magnetosphere using the DSX and Arase satellites. *Earth Planets Space* **2022**, *74*, 64. [\[CrossRef\]](#)
24. Lauben, D.; Linscott, I.; Inan, U. *The Broad Band Receiver (BBR) Instrument on the Demonstrations and Science eXperiments (DSX) Spacecraft*; Final Report no. AFRL-RV-PS-TR-2022-0114; Hanson Experimental Physics Laboratory, Stanford University: Stanford, CA, USA, 15 December 2022.
25. Neuffer, A.; Lunk, A. Diagnostic of a planar magnetron sputter source with Langmuir probes. In Proceedings of the International Symposium PLASMA '97 "Research and Applications of Plasmas", Jarnottówek near Opole, Poland, 10–12 June 1997; Invited Papers; Volume 2.
26. Galejs, J. Impedance of a Finite Insulated Cylindrical Antenna in a Cold Plasma with a Longitudinal Magnetic Field. *IEEE Trans. Antennas Propag.* **1966**, *14*, 727–736. [\[CrossRef\]](#)
27. Galejs, J. Impedance of a Finite Insulated Antenna in Cold Plasma with a Perpendicular Magnetic Field. *IEEE Trans. Antennas Propag.* **1966**, *14*, 737–748. [\[CrossRef\]](#)
28. Wunsch, D.A. Radiation from a strip of electric current in magnetoionic medium. *Can. J. Phys.* **1967**, *45*, 1675–1691. [\[CrossRef\]](#)
29. Bhat, B. Current distribution on an infinite tubular antenna immersed in a cold collisional magnetoplasma. *Radio Sci.* **1973**, *8*, 483–492. [\[CrossRef\]](#)
30. Seshadri, S.R. Input Admittance of a Cylindrical Antenna in Magneto-Ionic Medium. *J. Appl. Phys.* **1968**, *39*, 2407–2412. [\[CrossRef\]](#)
31. Song, P.; Reinisch, B.W.; Paznukhov, V.; Sales, G.; Cooke, D.; Tu, J.-N.; Huang, X.; Bibl, K.; Galkin, I. High-voltage antenna-plasma interaction in whistler wave transmission: Plasma sheath effects. *J. Geophys. Res.* **2007**, *112*, A03205. [\[CrossRef\]](#)
32. Soloviev, O.V. Low-frequency radio wave propagation in the Earth-ionosphere waveguide disturbed by a large-scale three-dimensional irregularity. *Radiophys. Quantum Electron.* **1998**, *41*, 392–402. [\[CrossRef\]](#)

33. Wait, J.R. Theory of Radiation From Sources Immersed in Anisotropic Media. *J. Res. Natl. Bur. Stand.—B. Math. Math. Phys.* **1964**, *68*, 119–136. [[CrossRef](#)]
34. Ginet, G.P. *VLF Far-Field Radiation from a Linear Dipole Antenna Immersed in a Plasma: Baseline Model for DSX*; Report no. 95-0002; Lincoln Laboratory, Massachusetts Institute of Technology: Lexington, MA, USA, 15 June 2020.
35. Bannister, P.R.; Harrison, J.K.; Rupp, C.C.; King, R.W.P.; Cosmo, M.L.; Lorenzini, E.C.; Dyer, C.J.; Grossi, M.D. Orbiting transmitter and antenna for spaceborne communications at ELF/VLF to submerged submarines. In Proceedings of the AGARD Conference Proceedings 529: ELF/VLF/LF Radio Propagation and Systems Aspects, Electromagnetic Wave Propagation Panel Symposium, Brussels, Belgium, 28 September–2 October 1992.
36. Poisel, R.A. *Antenna Systems and Electronic Warfare Applications*; Artech House: Boston, MA, USA; London, UK, 2012.
37. Modelski, J.; Jaszczyszyn, E.; Chaciński, H.; Majchrzak, P. *Measurements of Antenna Parameters*; OFPW, Warsaw University of Technology: Warsaw, Poland, 2004.
38. Bem, D.J. *Antennas and Propagation of Radio Waves*; WNT: Warsaw, Poland, 1973.
39. Celiński, Z. *Plasma*; PWN: Warsaw, Poland, 1980.
40. Bogdanov, A.F.; Vasin, V.V.; Dulin, V.N.; Ilin, V.A.; Krivitskiy, B.H.; Kuznetsov, V.A.; Labutin, V.K.; Molochkov, Y.B.; Piertsov, S.V.; Stiepanov, B.M.; et al. *Compendium on Radioelectronics*; Kulikovskiy, A.A., Ed.; WKiŁ: Warsaw, Poland, 1971.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

A Resource-Based Dynamic Pricing and Forced Forwarding Incentive Algorithm in Socially Aware Networking

Xuemin Zhang ^{1,2}, Yuan Li ^{1,2,*} , Zenggang Xiong ^{1,2,*} , Yanchao Liu ¹ , Shihui Wang ² and Delin Hou ^{1,2}¹ School of Computer and Information Science, Hubei Engineering University, Xiaogan 432000, China; zhxm9701@163.com (X.Z.); yanchaoliu@cuc.edu.cn (Y.L.); hdl00@stu.hubu.edu.cn (D.H.)² School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China; wsh@hubu.edu.cn

* Correspondence: 202221116012706@stu.hubu.edu.cn (Y.L.); jkxxzg2003@163.com (Z.X.)

Abstract: In socially aware networking, nodes typically behave selfishly due to resource constraints and social correlations, resulting in low network performance. To incentivize selfish nodes to actively participate in message forwarding, this paper proposes a resource-based dynamic pricing and forced forwarding incentive algorithm (DFIA). Firstly, the algorithm introduces virtual currency as a transaction medium and then designs a pricing function based on factors such as the node's resource status, participation contribution, location relevance, and social connectivity. It ensures that the forwarding service is transacted at a reasonable price through bargaining rules. Secondly, a forced forwarding strategy is implemented to compel selfish nodes, which are unwilling to participate in other nodes' message forwarding, to forward a certain number of non-local messages. Meanwhile, in order to prevent nodes from discarding messages and to ensure successful forwarding to the destination, specific rules are used to allocate contribution values to nodes that successfully participate in message forwarding. Lastly, to avoid false quotation behavior, blockchain technology is employed. Transaction information is packaged into blocks and added to the blockchain after consensus validation by other nodes in the network, ensuring the transparency and immutability of transaction data. Simulation results indicate that compared with the existing incentive algorithms, this algorithm not only enhances message delivery probability but also effectively reduces average latency.

Keywords: socially aware networking; selfish nodes; false quotation; virtual currency



Citation: Zhang, X.; Li, Y.; Xiong, Z.; Liu, Y.; Wang, S.; Hou, D. A Resource-Based Dynamic Pricing and Forced Forwarding Incentive Algorithm in Socially Aware Networking. *Electronics* **2024**, *13*, 3044. <https://doi.org/10.3390/electronics13153044>

Academic Editor: Domenico Rosaci

Received: 25 June 2024

Revised: 23 July 2024

Accepted: 30 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid development of network information technology, the mobile environment formed by smart devices (smartphones, tablets, in-vehicle devices) with the help of neighborhood-based wireless technologies (Bluetooth, WIFI) [1] has received more and more attention, and Socially Aware Networking (SAN) [2,3] has appeared accordingly. SAN research determines reasonable protocols to provide services for upper-layer applications by sensing and analyzing the contextual information of the mobile node [4] in the network. This is achieved through social network analysis and machine learning technologies, which analyze and learn from the behavior and environmental information of carriers obtained from intelligent mobile devices. From these data, social attributes such as community, interaction intensity, and mobility patterns are derived. By integrating these social attributes with a carrier behavior, effective routing protocols and message forwarding strategies are designed to provide high-quality services for upper layer applications. For instance, the studies in [5,6] made forwarding decisions based on users' social characteristics and similarities [7].

SAN (refer to Figure 1) leverages the social attributes of mobile carriers (i.e., subscribers) to design efficient network protocols. It represents a new approach in Delay Tolerant Network (DTN) [8] and Opportunistic Network (OppNet) [9]. In terms of operating principle, SAN is similar to DTN and OppNet [10]. On the one hand, due to the

dynamic topology caused by nodes' mobility within the network, there is a lack of stable end-to-end communication paths between the source and destination nodes. When the link between nodes is disrupted, the conventional routing methods fail to forward message data, and probabilistic encounters are often utilized to realize the forwarding in a multi-hop manner. On the other hand, during message forwarding, intermediate nodes typically use the "store-carry-forward" method to temporarily store messages [11–13], utilizing their short-range and low-cost devices to communicate with each other, thereby sharing data objects (such as pictures, advertisements, etc.) among nodes [7]. However, unlike DTN and OppNet, mobile devices in SAN are generally carried by objects such as humans, vehicles, and animals, which move to various places and engage in different activities, resulting in constantly changing network topology. Therefore, there are certain social attributes among nodes in SAN. It is extremely crucial to develop efficient solutions to motivate nodes to cooperate through these social attribute characteristics.

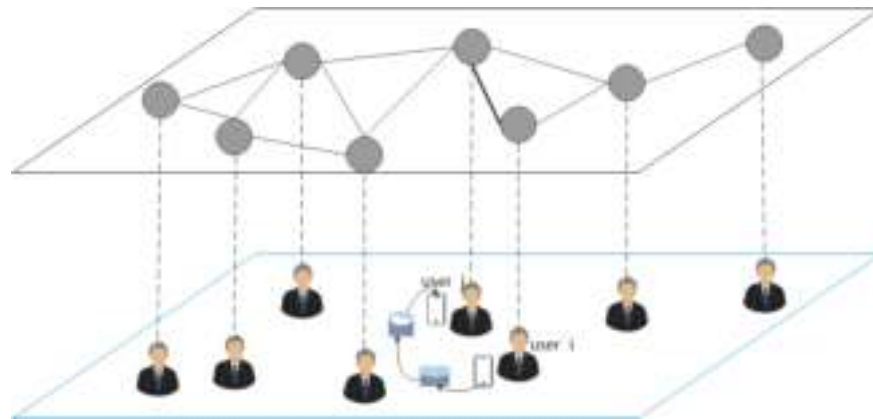


Figure 1. Socially aware networking.

As an emerging network model, SAN still has many numerous challenges for the research of node selfishness, message forwarding security issues, and other related technologies. In SAN, some nodes may exhibit selfish behaviors due to their limited resources resulting in poor delivery capability or poor social relationships with other nodes. These behaviors negatively impact the overall network performance and efficiency. The selfish behaviors of nodes are manifested as individual selfishness [14] and social selfishness [15]. Nodes exhibiting individual selfishness neither receive information from nor forward information to other nodes when their resources are scarce. Conversely, nodes displaying social selfishness only provide data forwarding services to familiar nodes and refuse to forward or receive data messages from nodes with poor social relationships. In SAN, in addition to the challenges posed by node selfishness, there also faces risks of malicious attacks such as tampering and stealing against forwarded messages and transaction records [16]. It is, thus, clear that effective incentive mechanisms and stringent security measures are needed to motivate nodes to participate in cooperation and prevent malicious attacks from occurring, which, in turn, ensures an efficient operation of the SAN and secure data transmission.

Currently, designing incentive mechanisms for nodes is the primary method to address the problem of node selfishness in SAN. Existing node cooperation incentive mechanisms can be categorized into three types: reputation-based incentive mechanisms, virtual currency-based incentive mechanisms, and game theory-based incentive mechanisms.

As described in paper [17], the authors accurately calculated reputation by combining the credibility and confidence of recommenders and fairly evaluated each recommendation. They employed a fair differential service mechanism to incentivize active participation. In [18], Roy et al. defined node reputation value and message forwarding cost. This strategy improves the message delivery efficiency by selecting nodes with a high reputation value and low cost in multicast data transmission as the next-hop nodes for message forwarding. Bounouni et al. [19] combined direct and indirect reputation to evaluate node trustworthi-

ness, allowing for nodes to accurately understand the status of neighboring nodes within their vicinity. Then, the nodes are classified into different cooperative states based on their reputation values, and corresponding incentives are implemented to encourage cooperation based on these states. These incentive mechanisms face challenges in monitoring, storing, and transmitting reputation values due to potential attacks by malicious nodes, which may cause inconsistencies in reputation values. Additionally, nodes near the set threshold may exhibit fluke mentality, maintaining just enough reputation value to avoid actively participating in forwarding.

Wang et al. [20] proposed an incentive evolutionary game model, which encouraged both normal and abnormal nodes to participate in network cooperation through the evolutionary process of survival of the fittest, ensuring a high level of cooperation and maintaining the continuous stability of the network. A repetitive game model (CGTFT) based on the cooperation evaluation function of neighboring nodes was established in the literature [21]. For rational nodes in the network, the cooperation degree evaluation function is designed by combining the residual energy and successful forwarding rate of the nodes. Then, the CGTFT model is used for the game, rewarding nodes that actively participate in cooperation and punishing selfish and uncooperative nodes. Additionally, a forgetting factor is introduced to facilitate the cooperation reconstruction for nodes that enter the punishment period. The study reported in [22,23] proposed a game theoretical pricing model based on the Rubinstein Bargain model, allowing for relay nodes to price their services according to residual energy and the value of the transmitted information, thus enhancing the distribution of message content within the network. These mechanisms employ game theory for analysis. However, they require high resource costs and are more suitable for symmetric networks, making them less friendly to asymmetric networks.

Xiao et al. [24] proposed an equivalent-exchange-based data forwarding incentive scheme (EEIS). EEIS combines the resource status and wealth status of nodes to price the messages that need to be forwarded and publicly disclose the wealth and resource status of both trading parties. In [25], Li et al. introduced a service pricing-based, two-stage incentive algorithm (STIA). STIA prices the forwarding service by considering service quality, message attributes, and resource status, and the authors proposed a two-stage incentive mechanism. It also formulates a currency overdraft scheme and a two-stage penalty mechanism to prevent nodes from defaulting after overdrawing currency. During service transaction, both nodes access a Credit Clearing Center (CCC) to understand each other's situation, preventing false quotation. However, using a third-party trusted institution to manage and prevent false quotation increases the risk of data leakage if there are security vulnerabilities or malicious attacks. Rao et al. [26] proposed a resource-constrained and Socially Selfish-Based Incentive Algorithm (RSIA). Nodes are rewarded with virtual currencies through bargaining and negotiation as an incentive for selfish nodes to be part of the cooperation. The algorithm introduces the Gini coefficient to evaluate the income balance of nodes, and it rewards or penalizes nodes based on their income balance. RSIA configures each node in the network with a trusted security module. Nodes only add the corresponding currency value when transaction information is verified as unmodified, thus avoiding false quotation. However, this algorithm does not take into account the positional relationship between nodes. When configuring the security module H for nodes to perform ACK validation, the reliance on manufacturers for node trust could compromise the entire network's security if a manufacturer's credibility was compromised. Therefore, ensuring the authenticity and accuracy of transactions and preventing false quotation remain critical issues to address.

In summary, regarding the above analysis, to incentivize selfish nodes to actively take part in message forwarding and prevent nodes from making false quotation, this paper proposes a resource-based dynamic pricing and forced forwarding incentive algorithm (DFIA). This algorithm analyzes the selfish behavior of nodes and considers their resource situation, contribution level, wealth status, and social relevance to negotiate message forwarding prices. Additionally, a forced forwarding strategy is introduced to compel

selfish nodes to forward non-local messages, thereby curbing the behavior of nodes that focus solely on their own interests and are unwilling to participate in network cooperation. To prevent false quotation, the transaction information is recorded on the blockchain, leveraging the decentralized and transparent nature of blockchain technology to ensure that prices cannot be tampered with.

The remainder of this paper is structured as follows: Section 2 introduces the network model and assumptions of the algorithm. Section 3 details the relevant definitions and implementation process of the proposed algorithm. Section 4 presents the simulation environment. We set Spray And Wait [27] as the underlying routing algorithm and combine the proposed DFIA, RSIA from paper [26], the STIA from paper [25], and the EEIS from paper [24] with the underlying routing algorithm for experimental comparison. The final part summarizes the entire paper and future outlook.

2. Network Model and Assumptions

This paper presents an abstract representation of the connections between nodes in Socially Aware Networking (SAN) as an undirected graph $G(D, E)$, where D is the set of all nodes and E represents the set of edges. Here, N denotes the number of nodes in SAN. The source node is the carrier of the message before transmission, and the relay node is the carrier that receives the message. Upon the meeting of the source node and the relay node, the formation of an edge between the nodes is initiated. $E(i, j)$ represents the edge between nodes i and j , where $i \neq j$.

Additionally, at the initial stage of the network simulation, this paper creates a genesis block. As transactions for message forwarding services between nodes are successfully completed, new transaction records are packaged into new blocks and sequentially linked after the genesis block. This ensures that all transaction records are securely and orderly stored in the blockchain formed from the genesis block. Nodes in the network collectively participate in and maintain a public, transparent, and tamper-proof blockchain.

According to the needs, this paper makes the following assumptions and definitions for the model:

1. Each node in the network has one and only one ID , and all nodes are rational, meaning they pursue the maximization of their economic benefits or other strategic goals while adhering to network protocols and rules.
2. The network comprises two distinct types of nodes: normal nodes and selfish nodes. Normal nodes participate in the storage, carrying, and forwarding of messages as expected; selfish nodes are those nodes that refuse to accept and forward any messages from other nodes due to low benefits or weak social connections.
3. Assume that the potential malicious behavior of nodes in SAN is limited to false quotation, rather than tampering with messages or collusion.
4. Only one node in the network is authorized to form a new block at any given time.

3. A Resource-Based Dynamic Pricing and Forced Forwarding Incentive Algorithm

In this paper, the forwarding of messages between nodes is treated as a transaction. The buyer is the carrier of the message before forwarding, and the seller is the carrier that receives the message. The forwarding service provided by the buyer is the commodity, and both parties use virtual currency as the transaction medium. Both parties offer an initial price based on a pricing function. If the bid is reasonable, the transaction is successful. Otherwise, the prices are adjusted according to specific bargaining rules, and the transaction is attempted again. If the transaction still fails, the next message is processed. Once a transaction is successful, the transaction information is written into a block and linked to the blockchain, ensuring transparency and immutability of the information.

3.1. Relevant Definitions

3.1.1. Delivery Urgency of Nodes

The delivery urgency of a node refers to the urgency with which a node can receive and forward messages, primarily determined by the node's remaining buffer space, remaining energy, and the remaining Time To Live (TTL) of the message. When all three are sufficient, it indicates that the node's delivery urgency is low, and the node has ample resources and time to forward and receive messages. As messages are continuously received and forwarded, the node's buffer space, its energy, and the message's TTL will inevitably decrease gradually, thereby increasing the node's delivery urgency. In this paper, we utilize a state index to represent this delivery urgency of a node.

The node's buffer state index is the ratio of the node's remaining buffer space at a given moment to the node's initial total buffer space, reflecting the node's buffering capability. The node's energy state index is the ratio of the node's remaining energy at a given moment to the node's initial total energy, reflecting the node's capability to forward messages. The message's TTL index is the ratio of the message's remaining TTL at a given moment to the message's initial TTL, which is one of the critical factors affecting the node's ability to deliver the message. Therefore, the delivery urgency $Q_{x,m}^h$ of node x at time h is defined as follows:

$$Q_{x,m}^h = \alpha * \left(1 - \frac{B_{r_x}}{B_{init_x}}\right) + \beta * \left(1 - \frac{E_{r_x}}{E_{init_x}}\right) + \gamma * \left(1 - \frac{T_{r_m}}{T_{init_m}}\right). \quad (1)$$

The weight values α , β , and γ are introduced to balance different optimization objectives. Since the problem involved is NP-Hard, the weight values between different instances will have a significant impact on the performance of the algorithm. Therefore, in our study, we adjusted different weight values through a large number of experiments, and then found the optimal weight configuration under a specific problem instance. B_{r_x} , E_{r_x} , and T_{r_m} represent the remaining buffer space, the remaining energy value, and the remaining TTL of the message m of the node x at the moment of h , respectively. B_{init_x} , E_{init_x} , and T_{init_m} represent the initial total buffer space, the initial total energy value, and the initial TTL of message m for node x , respectively.

3.1.2. Positional Relevance of Nodes

The positional relevance of nodes refers to the distance-related situation between the node and the message destination node. It is normalized using a Gaussian similarity function on the node's positional attribute function, serving as a measure of the node's positional attributes.

The primary location attributes of a node include the node's movement speed, the angle with the destination node, and the distance. The location attribute function $loc(x, d)$ of a node is defined as

$$loc(x, d) = \frac{v_x * |\cos A|}{D_{(x,d)}}, \quad (2)$$

where v_x denotes the moving speed of node x . Assume the coordinates of node x and destination node d are (a_x, b_x) and (a_d, b_d) , respectively. Then, $\cos A = \frac{a_x * a_d + b_x * b_d}{\sqrt{a_x^2 + b_x^2} * \sqrt{a_d^2 + b_d^2}}$, where $0 \leq A \leq \pi$, and A represents the angle between node x and the destination node. $D(x, d) = \sqrt{(a_x - a_d)^2 + (b_x - b_d)^2}$, where $D(x, d)$ represents the Euclidean distance between node x and the destination node.

The Gaussian similarity function [28] is used to normalize Equation (2), and the positional relevance of the nodes is calculated as

$$g(x) = e^{\frac{-(loc(x,d))^2}{2\sigma^2}}, \quad (3)$$

where $\sigma = 1$ is the scale parameter. As the location attribute function of node x gets smaller, the value of $g(x)$ approaches 1, indicating that the time for node x to reach the destination

node is shorter. In the pricing model, node x will be willing to pay a higher price for the forwarding service to quickly forward the message.

3.1.3. Participation Contribution of Nodes

In a network, the participation contribution of a node reflects its level of involvement in forwarding messages and its contribution to the overall network efficiency. This is a significant factor influencing the pricing of a node. A higher participation contribution is indicative of a node's willingness to provide services for other nodes, thereby increasing the delivery rate of messages within the network through its forwarding activities. Therefore, the DFIA algorithm introduces the cooperation degree and average contribution degree of nodes to measure their participation contribution during the message delivery process.

Whether a node is willing to cooperate is demonstrated by its willingness to consume its resources to forward messages generated by other nodes. The number of non-local messages forwarded by a node is directly proportional to its participation. Thus, the cooperation degree $X_x(h)$ of node x at time h is defined as

$$X_x(h) = \frac{F_x}{M_x}. \quad (4)$$

where F_x denotes the total number of non-local messages forwarded by node x . Each time node x successfully forwards a non-local message, then $F_x + 1$. M_x represents the number of encounters node x has with other nodes. Each time node x encounters another node: the $M_x + 1$.

When the message that a node helps forward is successfully delivered to the destination node, this paper assigns contribution values according to the node based on its position in the forwarding path to encourage active participation in network communication. The first relay node in the forwarding path is assigned the highest contribution value, as it determines the direction of message delivery. Additionally, the last relay node is given a relatively large contribution value because it completes the actual message delivery and is the task accomplisher. The other relay nodes in the forwarding path receive decreasing contribution values based on their positions in the forwarding path. The contribution value $C(i, n)$ assigned by node i on the n th path is represented as follows:

$$C(i, n) = \begin{cases} (1 - r)^{i-1} * r, & \text{if } 0 \leq i < L \\ (1 - r)^{i-1}, & \text{if } i = L \end{cases}, \quad (5)$$

$$avgC_x = \frac{\sum_{t=1}^n C_t(x, n)}{t}, \quad (6)$$

where r ($0 < r < 1$) denotes the contribution value allocation ratio factor, and L represents the total number of relay nodes on the path. Upon successful delivery of a message to the destination node, the contribution value of node x is accumulated in accordance with the aforementioned allocation scheme. Thereafter, the average contribution value of node x , $avgC_x$, is calculated according to Equation (6).

Therefore, the participation contribution PC_x of node x can be expressed by Equations (5) and (6) as

$$PC_x(h) = \rho * X_x(h) + (1 - \rho) * avgC_x, \quad (7)$$

where ρ is the weight factor by which $X_x(h)$ and $avgC_x$ have influence on PC_x and $\rho \in [0, 1]$.

3.1.4. Wealth Status of Nodes

Nodes in the network initially possess a specific amount of virtual currency, which reflects their forwarding capability to some extent. Based on the amount of virtual currency available to a node, it can be classified into three different virtual currency states: sufficient state, general state, and insufficient state. When $V_x(h) > V_{max}$, the node is in a sufficient

wealth state, at which time $C_x(h) > 1$. When $V_x(h) < V_{min}$, the node is in an insufficient wealth state, at which time $C_x(h) < 1$. When $V_{min} \leq V_x(h) \leq V_{max}$, the node is in a general wealth state, at which time $C_x(h) = 1$. The relevant calculation is as follows:

$$C_x(h) = \begin{cases} \frac{V_x(h)}{V_{max}}, & \text{Adequate State} \\ 1, & \text{General State} \\ \frac{V_x(h)}{V_{min}}, & \text{Inadequate State} \end{cases} \quad (8)$$

where $V_x(h)$ represents the amount of virtual currency owned by node x at moment h . The virtual currency thresholds V_{max} and V_{min} , respectively, represent the maximum and minimum amounts of virtual currency that may be owned by a node.

3.1.5. Degree of Social Correlation between Nodes

In SAN, the social correlations between nodes describe the extent to which some kind of connection or mutual influence exists between nodes. Social correlation is used to reflect the interaction and bonding between nodes in the network by analyzing the common neighbors and historical encounter records between nodes. Therefore, this paper comprehensively evaluates the social correlation among nodes from the dimensions of social overlap of nodes and the degree of contact between nodes.

The degree of overlap between two nodes is quantified by the number of common neighbors that they possess in SAN. These common-neighbor nodes are the bridge connecting two nodes, and they represent the similarity between nodes. Thus, the social $overlap_{xy}$ between node x and node y is defined as

$$overlap_{xy} = \frac{com_{xy}}{neigh_x + neigh_y - com_{xy}}, \quad (9)$$

where com_{xy} denotes the number of common-neighbor nodes between nodes x and y ; $neigh_x$ and $neigh_y$ represent the total number of neighbors of nodes x and y , respectively.

The degree of contact between nodes quantifies the intimacy of contact between these nodes through the analysis of the frequency and duration of encounters. If two nodes communicate with each other frequently, interact frequently and deeply, and the degree of contact between them is high, then, drawing on the definition of a friend in the real society, the two nodes are friends with each other. The contact degree $contact_{xy}$ between nodes x and y is calculated as follows:

$$contact_{xy} = \frac{Nmc_{xy} * Tm_{xy}}{N * accurentT}. \quad (10)$$

As mentioned above, Nmc_{xy} denotes the number of encounters between nodes x and y so far, Tm_{xy} denotes the sum of all encounter times between nodes x and y so far, N denotes the number of nodes in the current network, and $accurentT$ represents the current moment.

Based on Equations (9) and (10), the social correlation SC_{xy} between nodes is calculated as follows:

$$SC_{xy} = \varphi * overlap_{xy} + (1 - \varphi) * contact_{xy}, \quad (11)$$

where φ is the weight factor that determines the influence of $overlap_{xy}$ and $contact_{xy}$ on SC_{xy} , and $\varphi \in [0, 1]$.

3.2. Transaction Model

In SAN, message forwarding between encountering nodes is abstracted as a transaction process, where the forwarding service is bargained and traded using virtual currency. In this process, the node carrying the message before forwarding acts as the buyer, while the node receiving the message acts as the seller, and the message forwarding service acts as

the commodity. Both parties are required to disclose their wealth and resources as well as their offers for the forwarding service based on their respective considerations.

3.2.1. Buyer's Expenditure Function

To ensure that the message can be delivered to the next-hop node smoothly, buyer x must consider its own wealth reserve, delivery urgency, and message attributes carried, so as to determine a reasonable expenditure price as the compensation for providing forwarding services to the next-hop node. When x 's wealth reserve is sufficient, the buyer will synthesize its own delivery urgency and wealth reserve to give a reasonable price. When x 's wealth reserve is insufficient, with the passage of time, the survival time of the message will be gradually shortened, the node's remaining cache and energy will be gradually depleted, and the buyer's delivery urgency will become higher. Therefore, to relieve its pressure, the buyer will offer a higher transaction price. When x 's participation contribution is higher, indicating better service performance, it will benefit x in purchasing the forwarding service. If the node carries a larger message, the energy and cache consumed in forwarding will also increase, and x 's delivery urgency will also increase and, in turn, affect the bidding price for the message. In addition, x has to consider its own node position correlation. When its position correlation at the message destination node is high, it means that the time it takes for x to reach the destination node is shorter, and it will slightly raise the price in order to forward the message as quickly as possible. In summary, the expenditure function of buyer x to forward message m at time h is defined as

$$BN_{px}(h) = M_{size} * (Q_{(x,m)}^h + \sigma_1 * g(x)) * C_x(h) * PC_x(h). \quad (12)$$

where M_{size} denotes the size of m , $g(x)$ denotes the positional relevance of x , σ_1 is the weight occupied by the positional relevance of a node and satisfies $\alpha + \beta + \gamma + \sigma_1 = 1$, $C_x(h)$ denotes the wealth of x at time h , and $PC_x(h)$ denotes the participation contribution of node x at time h .

3.2.2. Seller's Income Function

When receiving messages carried by x , the seller y also needs to consider its own wealth reserve, delivery urgency, and the attributes of the carried message to price it, aiming to maximize its income. When y 's wealth reserve is sufficient, the node's willingness to provide message forwarding will decrease, so it has to combine with y 's delivery urgency to make reasonable pricing. When y 's wealth reserve is insufficient, it needs to obtain virtual currency from elsewhere at a lower price, and its willingness to provide message forwarding increases. Message forwarding is more favorable when y has a higher participation contribution. The size of the message is directly proportional to the energy and cache required for message forwarding; when the message is larger, the energy and cache consumption also increases for y providing a forwarding service, which, in turn, affects the pricing. In addition, y has to consider the social correlation between nodes. If the correlation between nodes is high, i.e., $SC_{xy} \geq SC_{xymin}$, then it indicates that the relationship between two nodes is better. Node y has a stronger willingness to forward the message, and the bidding price will be relatively low, as shown in Equation (13). On the contrary, if the correlation between nodes is low, i.e., $SC_{xy} < SC_{xymin}$, the forwarding node will bid relatively high, as shown in Equation (14).

$$SN_{py}(h) = M_{size} * (Q_{(y,m)}^h - \sigma_2 * (1 - SC_{xy})) * \frac{1}{C_y(h)} * (1 - PC_y(h)), \quad (13)$$

$$SN_{py}(h) = M_{size} * (Q_{(y,m)}^h + \sigma_2 * SC_{xy}) * \frac{1}{C_y(h)} * (1 - PC_y(h)). \quad (14)$$

where M_{size} denotes the size of m , SC_{xy} denotes the social correlation between x and y , and σ_2 denotes the weight value occupied by the social correlation between x and y , satisfying

$\alpha + \beta + \gamma + \sigma_2 = 1$. SC_{xymin} denotes the minimum threshold of social correlation between x and y , $C_y(h)$ denotes the wealth status of y at time h , and $PC_y(h)$ denotes the participation contribution of y at time h .

3.2.3. Forced Forwarding Strategy

Given the limited energy and cache resources available to nodes in the network, they tend to prioritize forwarding their own messages or refuse to forward non-local messages to ensure their generated messages are delivered successfully. This behavior greatly harms the overall interests of the network. To maintain the collective interests of the network and incentivize low-collaboration nodes to participate in cooperation, this paper introduces a forced forwarding strategy to restrict the behavior of those nodes that are only concerned about their own interests and are unwilling to cooperate in the network. This strategy promotes cooperation among nodes and optimizes network performance.

The size of the forced forwarding value CF_x of node x is determined by the collaboration degree $X_x(h)$ and contribution value $C(i, n)$ of node x , which is calculated as follows:

$$CF_x = \lceil 10 * (0.3 - X_x(h)) - C(i, n) \rceil. \quad (15)$$

Forcing the forwarding value to be updated automatically at a fixed frequency ensures that the forwarding behavior of the nodes can be monitored and regulated at certain time nodes. When $CF_x > 0$, node x must forward CF_x messages generated by other nodes before it can forward its own generated messages. When node x is restricted from forwarding, in order to forward its own messages smoothly, it can only improve the degree of collaboration by participating in forwarding messages generated by other nodes, and then gradually reduce its own forced forwarding value until it eventually restores the normal message forwarding capability.

3.2.4. Transaction Rules

In SAN, when the buyer x meets a seller y , both parties need to negotiate the pricing of the data forwarding service according to specific rules, and the algorithm's specific transaction rules are as follows:

(1) Both buyer and seller are fully informed of the message pricing and wealth status of the other party.

(2) After the transaction starts, the buyer's wealth is the price of expenditure given by buyer x as $BN_{px}(h)$ and the price given by seller y as $SN_{py}(h)$. If $BN_{px}(h) \geq SN_{py}(h)$, then message m is successfully forwarded, and the next message can be forwarded. If $BN_{px}(h) < SN_{py}(h)$, the first negotiation forwarding fails.

(3) If the first negotiation between buyer and seller fails, enter the second bargaining stage. By comparing the wealth of both parties, the respective prices are adjusted. The buyer adjusts $BN_{px}(h)$ by considering its cooperation degree, average contribution, and positional relevance, and the seller adjusts $SN_{py}(h)$ based on its cooperation degree, average contribution, and social correlation between nodes. The adjustment factors for the second negotiation are k_1 and k_2 , respectively. The second bargaining rules are as follows:

- If $C_x(h) > C_y(h)$, then $SN_{py}(h)$ remains unchanged, and $BN_{px}(h)$ is adjusted according to equation $BN_{px}(h) = BN_{px}(h) * (1 + k_1)$, where k_1 is calculated as follows:

$$k_1 = \omega_1 * X_x(h) + \omega_2 * avgC_x + \omega_3 * g(x), \quad (16)$$

where $X_x(h)$ denotes the cooperation degree of x , $avgC_x$ denotes the average contribution degree of x , and $g(x)$ denotes the positional relevance of node x . The weights ω_1 , ω_2 , and ω_3 denote the influence of cooperation degree, average contribution degree, and positional relevance of the node on the adjustment factor k_1 , respectively, and the sum of the three of them is 1. Here, the weights are dynamically adjusted using an adaptive weighting method, and the value of the weight is calculated using Equation (17):

$$\begin{cases} \omega_1 = \frac{X_x(h)}{X_{xh} + \text{avg}C_x + g(x)} \\ \omega_2 = \frac{\text{avg}C_x}{X_{xh} + \text{avg}C_x + g(x)} \\ \omega_3 = \frac{g(x)}{X_{xh} + \text{avg}C_x + g(x)} \end{cases} \quad (17)$$

- If $C_y(h) \geq C_x(h)$, then $BN_{px}(h)$ is unchanged, and $SN_{py}(h)$ is adjusted according to the formula $SN_{py}(h) = SN_{py}(h) * (1 - k_2)$, where k_2 is calculated as follows:

$$k_2 = \theta_1 * X_y(h) + \theta_2 * \text{avg}C_y + \theta_3 * SC_{xy}, \quad (18)$$

where $X_y(h)$ denotes the cooperation degree of y , $\text{avg}C_y$ denotes the average contribution degree of y , and SC_{xy} denotes the social correlation between nodes x and y . The weights θ_1 , θ_2 , and θ_3 represent the influence of collaboration degree, average contribution degree, and social correlation between nodes on the adjustment factor k_2 , respectively, and the sum of the three of them is 1. The dynamic adjustment of these weights also follows the same method as in Equation (17).

(4) If both nodes reach a consensus that $BN_{px}(h) \geq SN_{py}(h)$ after the second bargaining, the message will be forwarded as in step (2). If not, the transaction will be abandoned, and the next transaction will be continued. The message transaction flow is shown in Figure 2.

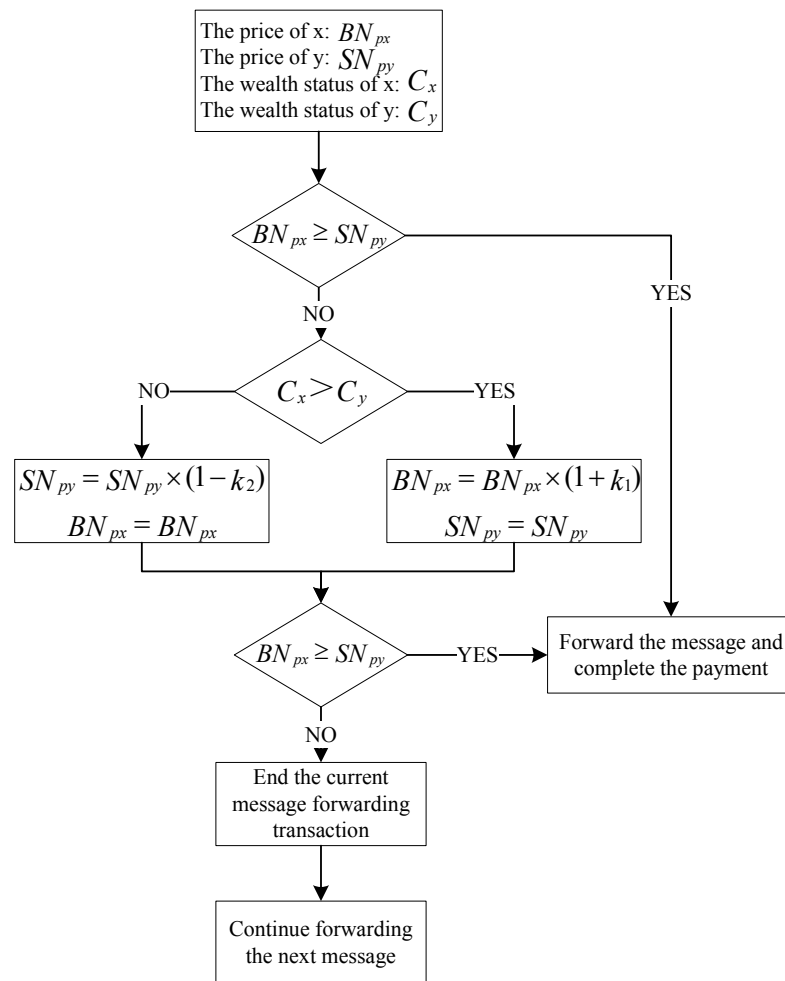


Figure 2. Flowchart of message transaction.

The detailed process of the normal message transaction algorithm is shown in Algorithm 1, which describes the complete bargaining process for the message forwarding

service when two nodes meet without a forced forwarding penalty. The detailed forwarding process is as follows: when nodes x and y encounter each other, x and y give their respective first bids for the forwarding service according to Equations (12)–(14), respectively. If $BN_{px} \geq SN_{py}$, then the message is forwarded, and the transaction price is set to the average of BN_{px} and SN_{py} . And then the transaction record is packed into the blockchain. On the contrary, according to the transaction rules described earlier, the price is adjusted based on the wealth status of x and y to make a new offer. If $BN_{px1} \geq SN_{py1}$, then one half of the sum of BN_{px1} and SN_{py1} is the transaction price. And the transaction record is packaged into the blockchain after the successful message forwarding. If the negotiation still fails after both parties re-adjust their prices, they give up forwarding this message m , and the next message is processed for forwarding.

Algorithm 1 Normal Message Transaction Algorithm

Input: $BN_{px}, SN_{py}, C_x, C_y$

Output: the successful forwarding message m

```

1: if  $BN_{px} \geq SN_{py}$  then
2:    $Cost_m = (BN_{px} + SN_{py}) / 2$ 
3:    $x$  sends message  $m$  to  $y$ ,  $y$  receives message  $m$ 
4:   if The amount of currency that  $x$  currently has is greater than  $Cost_m$  then
5:     Transaction successful ▷ createNewBlock
6:     The block is added to the end of the blockchain.
7:     Update the wealth of  $x$  and  $y$ 
8:     return  $m$ 
9:   end if
10: else
11:   Recalculate the cost of  $m$  and both parties' adjusted bids.
12:    $BN_{px1} = \text{getPriceTwice\_x}(BN_{px}, SN_{py}, C_x, C_y)$ 
13:    $SN_{py1} = \text{getPriceTwice\_y}(BN_{px}, SN_{py}, C_x, C_y)$ 
14:   if  $BN_{px1} \geq SN_{py1}$  then
15:      $Cost_m = (BN_{px1} + SN_{py1}) / 2$ 
16:      $x$  sends message  $m$  to  $y$ ,  $y$  receives message  $m$ 
17:     if The amount of currency that  $x$  currently has is greater than  $Cost_m$  then
18:       Transaction successful ▷ createNewBlock
19:       The block is added to the end of the blockchain.
20:       Update the wealth of  $x$  and  $y$ 
21:       return  $m$ 
22:     end if
23:   else
24:     Continue the next transaction.
25:   end if
26: end if

```

3.3. Message Forwarding Game Process

In SAN, the message forwarding transaction between nodes can be abstracted as a dynamic game process. The game participants in this process are the buyer x and the seller y . There are two main phases: whether the buyer x sends a message forwarding request and whether the seller y receives a message forwarding request at this time. The behavior of both parties in the two phases can either be cooperation or refusal of cooperation. It is to be expected that the rational nodes in the network select a strategy that maximizes their own benefits. In this paper, we use backward induction to analyze the two stages of the game process and demonstrate that mutual cooperation is the optimal strategy for both parties.

Suppose that at time h , for message m , the bidding price of x is $BN_{px}(h)$ and the bidding price of y is $SN_{py}(h)$, then the transaction price for the message is $Cost_m = \frac{BN_{px}(h) + SN_{py}(h)}{2}$.

(1) First, analyze the second stage of the game process, whether the seller y receives the message forwarding request or not, and its behavior determines the node's profit. When y accepts a message forwarding request from x , i.e., chooses a cooperative behavior, the current gain E_y of y is calculated as follows:

$$E_y = SN_{py}(h) - Cost_m. \quad (19)$$

When y refuses to receive a message forwarding request from x , i.e., chooses to refuse a cooperative behavior, there is no gain. In addition, since y did not accept the message while encountering x , the number of non-local messages forwarded by y is less, while the number of encounters increases and the degree of collaboration of y decreases, resulting in the participation of the contribution also decreasing. From Equations (13) and (14), it can be seen that y 's next bid for other messages becomes higher, which may lead to the failure of the transaction. In contrast, y will be willing to receive messages.

(2) Next, analyze whether the buyer x sends message forwarding requests or not. If x sends message forwarding requests at the time h , i.e., chooses the cooperative behavior, then the currency it saves E_x can be expressed as

$$E_x = BN_{px}(h) - Cost_m, \quad (20)$$

If x does not send a message forwarding request at time h , i.e., it refuses to cooperate, first, analyze the cost that x has to pay from the point of view of the message to be forwarded. At time $h + 1$, the survival time of the message passes with it, and in order to successfully forward m , the extra cost H_1 that x has to pay can be expressed as

$$H_1 = M_{size} * C_x(h + 1) * PC_x(h + 1) * \varphi_1 * \frac{T_{re}(h) - T_{re}(h + 1)}{TTL}, \quad (21)$$

where $T_{re}(h + 1)$ denotes the remaining TTL of m at time $h + 1$, and the meanings of the other symbols correspond to those in Equation (12).

Additionally, the cost to be paid by x is analyzed in terms of buyer x 's energy and cache. Since the energy of x is strictly decreasing as time increases, and x refuses to cooperate, its cache is also partially occupied by the message m . Therefore, at time $h + 1$, if x wants to send a forwarding request, and its proposed price will also increase, then the additional price H_2 that x has to pay can be expressed as

$$H_2 = \varphi_2 * \left(\frac{M_{size}}{B_{rx}(h + 1)} + \frac{M_{size}}{E_{rx}(h + 1)} \right) * L_x * C_x(h + 1) * PC_x(h + 1). \quad (22)$$

where L_x denotes the length of the message to be forwarded by x at time $h + 1$, $B_{rx}(h + 1)$ denotes the remaining buffer ratio of x at time $h + 1$, $E_{rx}(h + 1)$ is the remaining energy ratio of x at time $h + 1$, and the meanings of the other symbols match those in Equation (12).

In summary, the additional cost H to be paid by buyer x for choosing to refuse cooperation at time h is at least

$$H = \min\{H_1, H_2\}. \quad (23)$$

Obviously, $E_x > 0 > -H$, so x must choose to send a message forwarding request.

After the above analysis, both buyers and sellers will choose the cooperative behavior under the premise of both obtaining the maximum benefit, i.e., (cooperation, cooperation) is the optimal strategy.

3.4. Security Module

This paper adopts the incentive strategy based on virtual currency to design the algorithm, so it is inevitable that the behavior of nodes' false quotation will occur in the process of each transaction, which seriously undermines the normal order of the transaction.

In order to prevent nodes from illegally tampering with the amount of currency, this paper uses blockchain technology to create a security module to guarantee open and transparent message forwarding transactions between nodes. Blockchain is a decentralized digital ledger for economic transactions with the advantages of transparency and security [29–32]. In the network, records of all successful transactions are recorded in a blockchain structure, usually in a sequential order, and cryptographically guaranteed to be non-tampering and unforgeable. Each node in the chain is independent and equal, relying on a consensus mechanism to ensure data consistency, with the characteristics of decentralization, openness and transparency, and non-tampering [33].

3.4.1. Module Creation

When storing transaction data, each node in the network generates a public key PK and a secret key SK by means of a public key encryption algorithm (e.g., RSA). Among them, the SK is kept securely by the node, while the PK is publicly available on the network. To ensure the security of the transaction process and prevent false quotations, this paper performs the following process for the transaction record generated between the initiator A and the acceptor B of a transaction, which contains the transaction parties A and B , the transaction price, cost, etc.:

(1) Block creation: When the message transaction between A and B is successfully completed, use the hash function (SHA-256) to calculate the hash value of each item in the transaction record. Then, A 's SK is used to sign the hash value $H(cost)$ of the price of the transaction, generating $Sig_{SK}(H(cost))$. Then, the transaction record, the hash value, and the signature are packed into a new block and broadcast to the whole network so that other nodes can verify.

(2) Verification: B can use A 's PK to verify the digital signature and recalculate the hash value $H'(cost)$ of the transaction price. If these two hash values match, then the verification process is successful, which indicates that the transaction data have not been tampered with in terms of price during transmission. Other nodes in the network will check whether the transaction in the block is valid after receiving the new block.

(3) Consensus achievement: When a sufficient number of nodes confirm that the transaction is valid and a consensus is reached, the block is considered valid.

(4) Blockchain update: Once a new block has gained network-wide consensus, it is added to the end of the blockchain as shown in Figure 3. This update process ensures consensus and consistency of transaction history across the network.



Figure 3. Schematic diagram of the blockchain.

The tampering of transaction data is ensured by the use of hash algorithms in the modules created through the above process. Each new block contains the hash value of the previous block, and any tampering with the transaction data will change its hash value, thus disrupting the continuity of the entire blockchain and making any dishonest behavior easy to be recognized and rejected by other nodes in the network. Moreover, the transaction data are transparent and trustworthy due to the fact that the verified transaction records are written in blocks and added to the blockchain, which, in turn, achieves transparency and trustworthiness of the transaction data. Any node participating in the network can access and verify these transaction records, again effectively preventing the occurrence of a

false quotation behavior. All nodes follow the longest chain principle, meaning that when multiple competing branch chains exist, nodes will choose to join the longest valid chain as the main chain, ensuring the robustness and security of the blockchain. Thus, the fairness and security of the message transaction process in the network is ensured through the use of the blockchain.

3.4.2. Forwarding Scheme

The complete message forwarding service process in DFIA is shown in Algorithm 2. Suppose node x meets node y , x needs to forward message m to y , where CF_x denotes the forced forwarding value of x . CF_x is judged before the message forwarding transaction. When $CF_x > 0$, x enters into the forced forwarding phase; it is forced to forward non-local messages. For each successful message forwarding service transaction, the value of CF_x is decremented by one, and vice versa; it continues to forward the next message. When $CF_x \leq 0$, x carries out the normal message forwarding transaction according to Algorithm 1. During the message forwarding service transaction, the forwarding service is priced according to Equations (12)–(14), and bargaining is carried out according to specific transaction rules. If the bargaining is successful and the message is forwarded successfully, the transaction information including both parties x and y , transaction price, etc., is packed into the block and added to the blockchain after verification. Otherwise, the transaction fails and the next message is forwarded.

Algorithm 2 DFIA Algorithm

Input: CF_x, x, y
Output: the successful forwarding message m

```

1: if  $CF_x > 0$  then
2:   if The source node of the message is not the node  $x$  then
3:     if  $BN_{px} \geq SN_{py}$  then
4:        $Cost_m = (BN_{px} + SN_{py})/2$ 
5:       if The amount of currency that  $x$  currently has is greater than  $Cost_m$  then
6:         Transaction successful                                ▷ createNewBlock
7:         The block is added to the end of the blockchain.
8:         Update the wealth of  $x$  and  $y$ 
9:          $CF_x = CF_x - 1$ 
10:        return  $m$ 
11:      end if
12:    else
13:      Recalculate the cost of  $m$  and both parties' adjusted bids.
14:       $BN_{px1} = \text{getPriceTwice}_x(BN_{px}, SN_{py}, C_x, C_y)$ 
15:       $SN_{py1} = \text{getPriceTwice}_y(BN_{px}, SN_{py}, C_x, C_y)$ 
16:      if  $BN_{px1} \geq SN_{py1}$  then
17:         $Cost_m = (BN_{px1} + SN_{py1})/2$ 
18:        if The amount of currency that  $x$  currently has is greater than  $Cost_m$  then
19:          Transaction successful                                ▷ createNewBlock
20:          The block is added to the end of the blockchain.
21:          Update the wealth of  $x$  and  $y$ 
22:           $CF_x = CF_x - 1$ 
23:          return  $m$ 
24:        end if
25:      else
26:        Continue the next transaction
27:      end if
28:    end if
29:  end if
30: else
31:   ▷ Normal Message Transaction Algorithm
32: end if
33: function CREATENEWBLOCK( $hostX, other, m, cost$ )
34:    $previousHash \leftarrow \text{blockchain.getLastBlock().getHash}()$ 
35:    $timestamp \leftarrow \text{SimClock.getTime}()$ 
36:   Sign the hash value of the transaction cost using  $A$ 's private key  $SK$ .
37:    $newBlock \leftarrow \text{new Block}(hostX, other, m, cost, previousHash, timestamp, 0)$ 
38:    $blockHash \leftarrow newBlock.calculateHash()$ 
39:    $newBlock.setHash(blockHash)$ 
40:   return  $newBlock$ 
41: end function

```

4. Performance Evaluation

4.1. Simulation Environment and Scenario

The experiments reported in this chapter were conducted on a personal computer with a CPU of 3.60 GHz, 16.0 GB of RAM, and Windows 11 operating system, running the Opportunistic Network Environment simulator [34] (ONE) through the IntelliJ IDEA 2021.2.1 software operation. In the experiments, this study adopted Spray and Wait routing as the underlying routing algorithm and compared DFIA with RSIA in [26], STIA in [25], and EEIS in [24] for simulation. The following five simulation scenarios were designed:

- (1) S + selfish nodes;
- (2) S + selfish nodes + DFIA;
- (3) S + selfish nodes + RSIA;
- (4) S + selfish nodes + STIA.
- (5) S + selfish nodes + EEIS.

The specific experimental parameter settings are presented in Table 1.

Table 1. Parameter settings for simulation experiments.

Parameter	Value
Simulation time	12 h
Number of nodes	180
Simulation area	4500 m × 3400 m
Transmit speed	250 kB/s
Transmit range	10 m
Movement model	ShortestPathMapBasedMovement
Routing protocol	Spray and Wait
Buffer size	10 M
Message TTL	6 h
Transmit energy	4 W
Initial energy	2000 J
Message creation interval	25 s–35 s
Message Size	500 kB–1 MB
α	0.1
$\beta, \gamma, \sigma_1, \sigma_2$	0.3
ρ	0.6
φ	0.45

4.2. Performance Parameters

The simulation experiments mainly use the following three parameters as the evaluation criteria to assess the effectiveness and practicality of the incentive algorithm.

Message delivery probability: this is the ratio of the total number of messages successfully delivered to the destination node to the total number of messages created. A higher message delivery rate indicates a greater success rate of the node in forwarding the message to the destination node. Message delivery probability is an extremely important parameter for evaluating the network performance.

Average latency: this is the average time consumed by a message from its inception to successful forwarding to the destination node. Lower latency means that less network resources will be consumed during message forwarding, and the message will take less time to reach the destination.

Network overhead: this is represented by the ratio of the difference between the number of messages relayed by nodes in the network and the number of successfully forwarded messages to the number of successfully forwarded messages. A reduction in network overhead is indicative of a reduction in energy consumption by the node during message forwarding.

4.3. Simulation Results

In this study, evaluation experiments were designed in terms of different selfish node percentages and different TTLs. In the first evaluation, we observed the experimental results by setting the percentage of selfish nodes to 0%, 20%, 40%, 60%, 80%, and 100%. Figures 4–6 illustrate the comparative performance of each algorithm in terms of message delivery probability, average latency, and network overhead in three dimensions when different percentages of selfish nodes exist in the above five scenarios. In the second evaluation, the percentage of selfish nodes was set to 60%, and the message TTL was set to 2 h, 4 h, 6 h, 8 h, and 10 h, respectively. Figures 7–9 show the changes in message delivery probability, average latency, and network overhead of each algorithm for the five scenarios mentioned above under different message TTLs. A total of 30 trials were conducted for each set of evaluation experiments, and the mean value of the data from these trials was taken as the final experimental result.

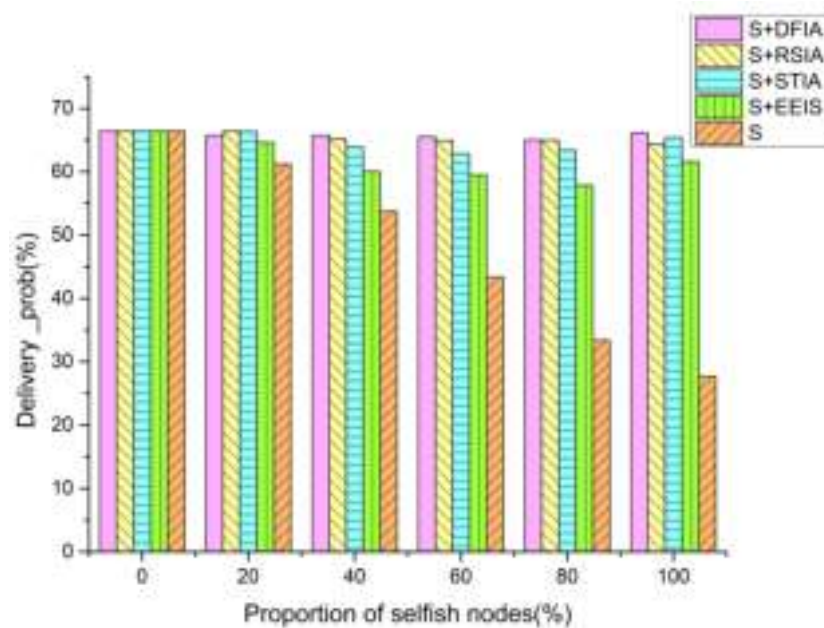


Figure 4. Message delivery probability with different percentages of selfish node.

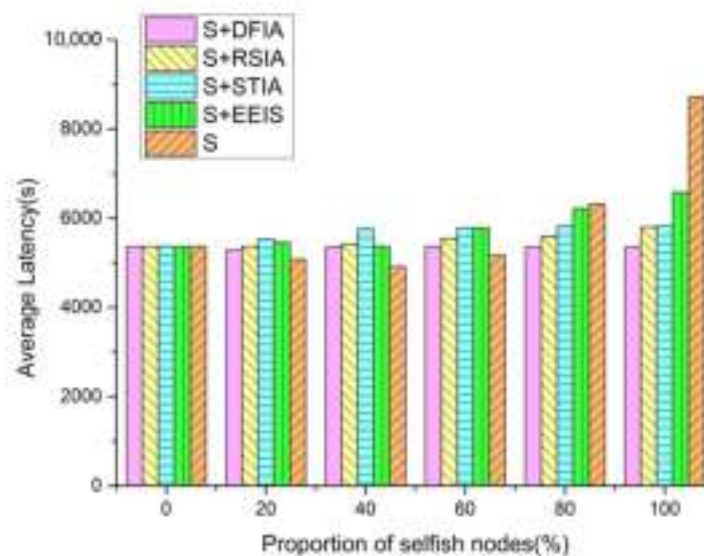


Figure 5. Average latency with different percentages of selfish nodes.

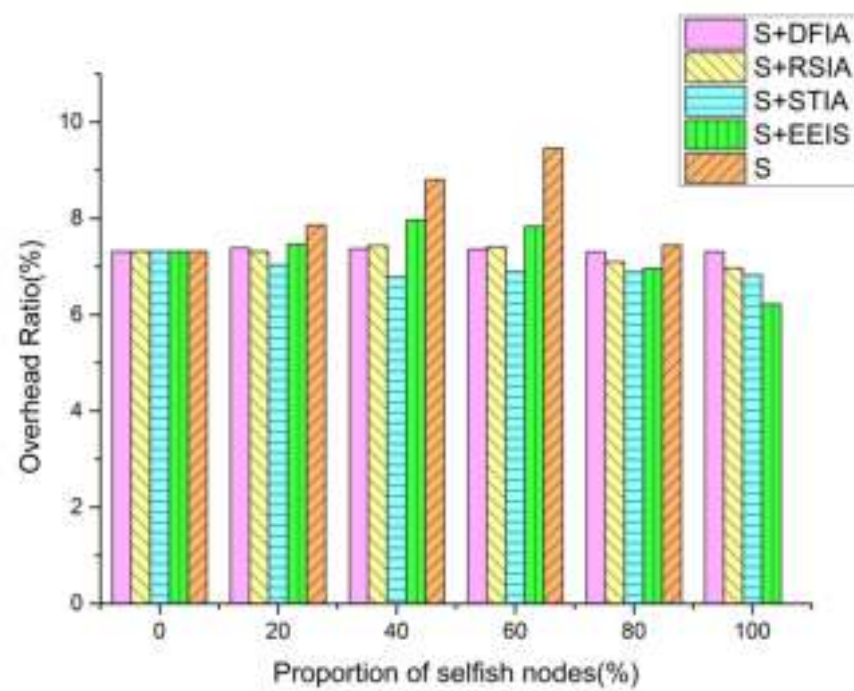


Figure 6. Network overhead for different percentages of selfish nodes.

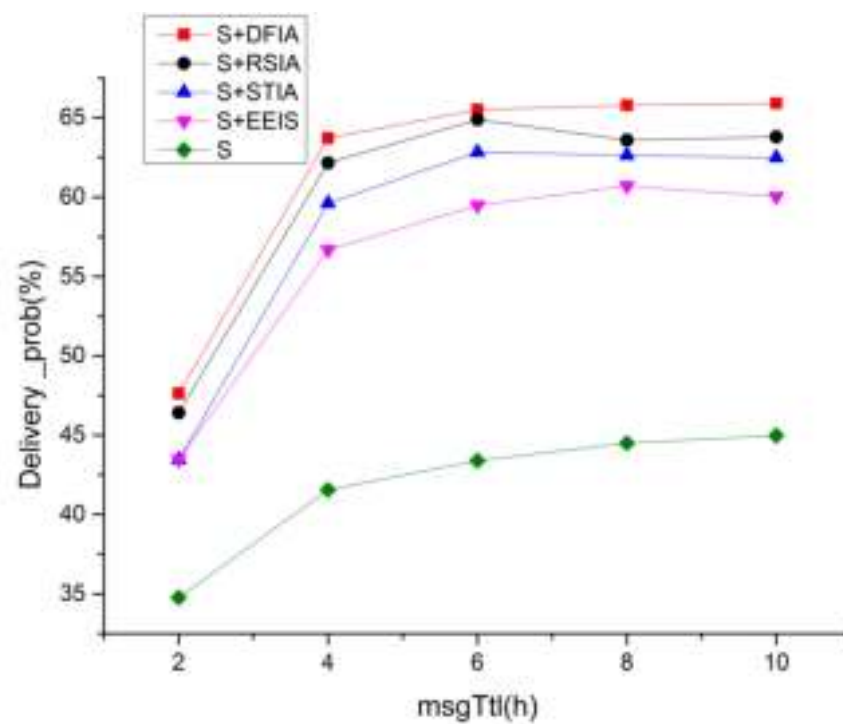


Figure 7. Message delivery probability under different message TTLs.

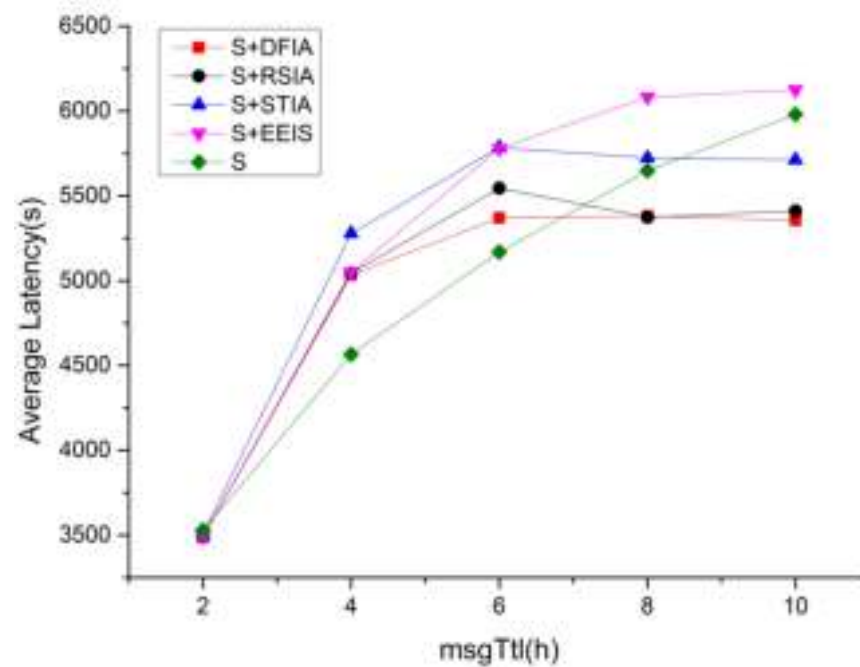


Figure 8. Average latency at different message TTLs.

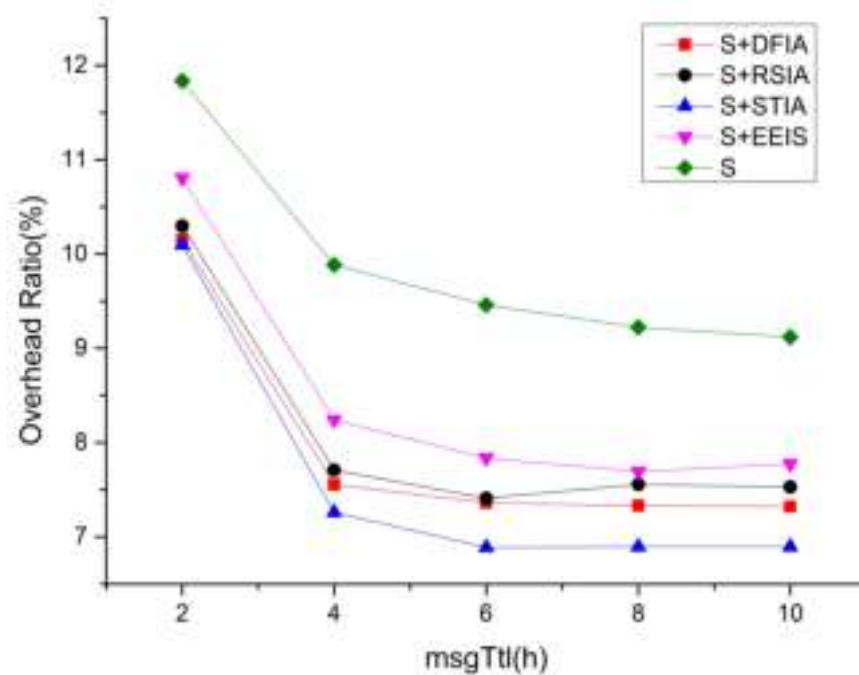


Figure 9. Network overhead at different message TTLs.

4.3.1. Different Selfish Node Percentages

As shown in Figure 4, the probability of message delivery for the five scenarios with different selfish node percentages is depicted. The decrease in message delivery percentages can be attributed to the fact that selfish nodes do not participate in the forwarding cooperative behavior, which reduces the success rate of messages reaching the destination node. The overall trend indicates that the addition of selfish nodes results in DFIA, RSIA, and STIA acting as an incentive in the case of nodes with a selfish behavior. However, DFIA is more effective compared to RSIA and STIA. S has a more pronounced trend of decreasing message delivery rate as it does not use any incentives. RSIA focuses on the resource constraints and possible presence of high affluent nodes' resource constraints

and the possible impact of preferred selfishness on the network. STIA takes into account the quality of forwarding service of the nodes as well as the possible effects of short-term currency shortage in making decisions about nodes' forwarding behaviors. EEIS uses node resource pricing and adopts the mode of equivalent exchange to carry out transactions. However, they do not consider the influence of location correlation between nodes and the influence of strong and weak social correlations between nodes. DFIA comprehensively considers the characteristics of node location and social correlations. Although its message delivery probability is 0.68% lower than RSIA when there are 20% selfish nodes, with the continuous increase in selfish nodes, the delivery probability of DFIA is higher than other algorithms. Based on the evaluation of the overall effect of the experiment, DFIA demonstrates the optimal performance in terms of message delivery probability.

Figure 5 illustrates the average latency for the five scenarios with different proportions of selfish nodes. The increase in the average delay means that the addition of selfish nodes results in a longer time for message forwarding to the destination. The data presented in Figure 5 reveal that as the proportion of selfish nodes increases from 0% to 100%, the average latency increases in all scenarios, with varying degrees of increase. S has the most significant increase in average latency, especially when the proportion of selfish nodes reaches 60%; its average latency value increases dramatically because no incentive measures are adopted. In the remaining four scenarios, the increase in average latency is relatively smooth and does not appear to rise sharply. Among them, the average latency of DFIA is the most stable, and its average latency is lower than that of RSIA, STIA, and EEIS. This indicates that the adoption of DFIA incentive strategy is more conducive to mitigating the impact of selfish nodes, and it improves the delivery probability of the message and reduces the average latency at the same time.

Figure 6 depicts the network overhead in the five scenarios under different percentages of selfish nodes. From the data in the figure, it can be seen that with the increasing proportion of selfish nodes, the network overhead of the other four scenarios, except for S with no incentives, is generally decreasing and changing more stably. In the S scenario, the network overhead decreases sharply when the proportion of selfish nodes exceeds 60%, and it directly decreases to 0 when the proportion of selfish nodes in the network is 100% because there are more and more selfish nodes in the network, and the nodes are not able to relay the messages or successfully deliver the messages to the destination node in the absence of incentive strategies. When the proportion of selfish nodes increases to 60%, the overall network overhead of DFIA is high compared to RSIA, STIA, EEIS, and S with no incentives. This is because DFIA introduces blockchain technology in the use of RSA to encrypt the transaction records, leading to a higher overall network overhead at the expense of higher network overhead in increasing the message delivery probability and ensuring the security of the transaction.

4.3.2. Different TTLs

Figure 7 illustrates the message delivery probability for the above scenarios under different message TTLs. It is evident that the message delivery probability increases with each increase in message TTL. The increase in message TTL means that the nodes are not restricted by too many time constraints when forwarding messages. There is enough time to forward the messages, which can avoid the situation of dropping messages and the resulting lower performance of the network. Compared to the other four algorithms, the message delivery probability of DFIA is clearly optimal, which depends on the fact that DFIA takes into account the factors leading to the selfishness of nodes in many aspects, which, in turn, motivates the selfish nodes to take part in the cooperation. In contrast, RSIA, STIA, and EEIS do not consider the effect of the location relevance of nodes on nodes and do not distinguish the strength of the social correlation obviously, leading to less effective incentives. It can be seen that DFIA can effectively prevent nodes from dropping messages, making the message delivery probability optimal.

Figure 8 illustrates the average latency of the four scenarios under different message TTLs. It is clear that the average latency increases for all of them as the message TTL increases. An increase in the message TTL means that the messages have a greater chance of being delivered to a greater distance in the network, which, in turn, leads to an increase in the number of packets in the nodes congestion when data forwarding is performed and a consequent increase in the latency of the network. The average delay of these five scenarios has a significant increasing trend with the increase in the message TTL, and the maximum increase in average delay is observed for S, which does not take any incentives. Comparatively, the DFIA, RSIA, STIA, and EEIS have a smaller increase in average latency. Among them, the average latency of DFIA is lower compared to RSIA, STIA, and EEIS, which is due to the fact that although STIA considers the node resources and quality of service, it ignores the node's social correlation factor, which increases the average latency of the message to a certain extent. RSIA only considers the selfish reasons of nodes, but does not consider the positional relationships between them and distinguishes between strong and weak social correlations, which leads to a higher average latency. EEIS adopts quadratic bargaining and does not consider that the message cannot be transmitted in time due to node packet loss, which affects the average latency. DFIA considers the positional relationships between nodes and introduces the social correlation factor, thereby enhancing the delivery rate of the message and reducing the average latency.

Figure 9 illustrates the comparison of network overhead. As the message TTL increases, the network overhead in all scenarios gradually decreases. Since S does not take any incentive measures, its network overhead is higher than that of DFIA, RSIA, STIA, and EEIS. STIA considers the quality of service of the nodes and reduces the message interactions during the bargaining process, thereby optimizing encounter time for message delivery and reducing network overhead. EEIS only considers node resource pricing and uses credit value for monetary lending, and it does not consider the packet loss behavior caused by the decline in transaction capacity due to excessive borrowing and failure to repay, which leads to a high network overhead. DFIA considers node resources, location attributes, and social correlations for the bargaining game, which makes its incentive effect more significant than that of RSIA, and it also has a smaller network overhead. Due to the consideration of multiple factors during bargaining in DFIA, more information interactions are generated, and the network overhead is higher compared to STIA. But DFIA increases the message delivery probability and reduces the average latency, and the overall incentive effect is better than that of STIA.

5. Conclusions

In this paper, we propose a resource-based dynamic pricing and forced forwarding incentive algorithm (DFIA). DFIA bargains the forwarding service required by nodes by integrating the delivery urgency determined by the node's resource status, location relevance, and distinguishing between strong and weak social correlations. During the transaction process, the forced forwarding strategy is designed to improve the degree of the nodes' collaboration. In order to prevent nodes from a false quotation behavior, DFIA uses blockchain technology for encryption to achieve transparency and credibility of transaction data. Simulation experiment results show that DFIA reduces the average latency while increasing the message delivery probability and effectively incentivizes selfish nodes to participate in cooperation. However, the algorithm uses RSA to encrypt the transaction records resulting in a higher overall network overhead, which needs to be addressed and optimized in future work. Next, we will delve into the security measures that should be taken for possible malicious behaviors such as nodes colluding with each other and tampering with messages in a SAN.

Author Contributions: Conceptualization, X.Z., Y.L. (Yuan Li) and Z.X.; methodology, Y.L. (Yuan Li); software, Y.L. (Yuan Li); validation, Z.X., Y.L. (Yuan Li) and X.Z.; resources, X.Z., Y.L. (Yuan Li) and Z.X.; data curation, Y.L. (Yuan Li); writing—original draft preparation, Y.L. (Yuan Li) and D.H.; writing—review and editing, Z.X., Y.L. (Yanchao Liu) and S.W.; project administration, Z.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant number NO. 61972136) and the Hubei Provincial Department of Education Outstanding Youth Scientific Innovation Team Support Foundation (grant number NOs. T201410, T2020017).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Wang, Z.; Liao, J.; Cao, Q.; Qi, H.; Wang, Z. Achieving k-Barrier Coverage in Hybrid Directional Sensor Networks. *IEEE Trans. Mob. Comput.* **2014**, *13*, 1443–1455. [\[CrossRef\]](#)
- Tsugawa, S.; Ohsaki, H. Community Structure and Interaction Locality in Social Networks. *J. Inf. Process.* **2015**, *23*, 402–410. [\[CrossRef\]](#)
- Ning, Z.; Liu, L.; Xia, F.; Jedari, B.; Lee, I.; Zhang, W. CAIS: A Copy Adjustable Incentive Scheme in Community-Based Socially Aware Networking. *IEEE Trans. Veh. Technol.* **2017**, *66*, 3406–3419. [\[CrossRef\]](#)
- Xu, F.; Xu, Q.; Xiong, Z.; Xiao, N.; Xie, Y.; Deng, M.; Hao, H. Intelligent distributed routing scheme based on social similarity for mobile social networks. *Future Gener. Comput. Syst.* **2019**, *96*, 472–480. [\[CrossRef\]](#)
- Zhu, K.; Li, W.; Fu, X. SMART: A Social- and Mobile-Aware Routing Strategy for Disruption-Tolerant Networks. *IEEE Trans. Veh. Technol.* **2014**, *63*, 3423–3434. [\[CrossRef\]](#)
- Annalisa, S.; Antonio, C.; Floriano, D.R.; Peppino, F. Routing in mobile opportunistic social networks with selfish nodes. *Wirel. Commun. Mob. Comput.* **2019**, *2019*, 6359806.
- Xia, F.; Liu, L.; Jedari, B.; Das, S.K. PIS: A Multi-Dimensional Routing Protocol for Socially-Aware Networking. *IEEE Trans. Mob. Comput.* **2016**, *15*, 2825–2836. [\[CrossRef\]](#)
- Galati, A. *Delay Tolerant Network*; LAP Lambert Academic Publishing: Koln, Germany, 2010.
- Luciana, P.; Andrea, P.; Marco, C. Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks. *IEEE Commun. Mag.* **2006**, *44*, 134–141.
- Xia, F.; Liu, L.; Li, J.; Mohammed, A.A.; Tianruo, Y.L.; Jianhua, M. BEEINFO: Interest-Based Forwarding Using Artificial Bee Colony for Socially Aware Networking. *IEEE Trans. Veh. Technol.* **2015**, *64*, 1188–1200. [\[CrossRef\]](#)
- Wu, J.; Yu, G.; Guan, P. Interest Characteristic Probability Predicted Method in Social Opportunistic Networks. *IEEE Access* **2019**, *7*, 59002–59012. [\[CrossRef\]](#)
- Elnaz, N.; Nahideh, D. FTR: Features tree based routing in mobile social networks. *Wirel. Netw.* **2020**, *26*, 283–291.
- Mao, Y.; Zhou, C.; Qi, J.; Zhu, X. A fair credit-based incentive mechanism for routing in DTN-based sensor network with nodes' selfishness. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 232. [\[CrossRef\]](#)
- Loudari, S.E.; Benamar, N. Effects of Selfishness on the Energy Consumption in Opportunistic Networks: A Performance Assessment. In Proceedings of the 2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS), Fez, Morocco, 3–4 April 2019; pp. 1–7.
- Li, Q.; Gao, W.; Zhu, S.; Cao, G. A routing protocol for socially selfish delay tolerant networks. *Ad Hoc Netw.* **2012**, *10*, 1619–1632. [\[CrossRef\]](#)
- Zhang, X.; Deng, H.; Xiong, Z.; Liu, Y.; Rao, Y.; Lyu, Y.; Li, Y.; Hou, D.; Li, Y. Secure Routing Strategy Based on Attribute-Based Trust Access Control in Social-Aware Networks. *J. Signal Process. Syst.* **2024**, *96*, 153–168. [\[CrossRef\]](#)
- Chang, J.; Pang, Z.; Xu, W.; Wang, H.; Yin, G. An incentive compatible reputation mechanism for P2P systems. *J. Supercomput.* **2014**, *69*, 1382–1409. [\[CrossRef\]](#)
- Roy, A.; Acharya, T.; Das, S. Social-based reputation-aware data forwarding for improved multicast delivery in the presence of selfish nodes in DTNs. *Int. J. Commun. Syst.* **2020**, *33*, e4235. [\[CrossRef\]](#)
- Bounouni, M.; Bouallouche-Medjkoune, L. Acknowledgment-based punishment and stimulation scheme for mobile ad hoc network. *J. Supercomput.* **2018**, *74*, 5373–5398. [\[CrossRef\]](#)
- Wang, E.K.; Chen, C.M.; Yiu, S.M.; Hassan, M.M.; Alrubaiyan, M.; Fortino, G. Incentive evolutionary game model for opportunistic social networks. *Future Gener. Comput. Syst.* **2020**, *102*, 14–29. [\[CrossRef\]](#)
- Wu, J.; Mo, W.; Yin, X.; Bai, G. Research on opportunistic routing incentive mechanism based on repeated game in WMNs. *Comput. Eng. Appl.* **2018**, *54*, 5.
- Haq, A.; Faheem, Y. A peer-to-peer communication based content distribution protocol for incentive-aware delay tolerant networks. *Wirel. Netw.* **2020**, *26*, 583–601. [\[CrossRef\]](#)
- Li, L.; Zhong, X.; Jiang, Y. IAR-GT: An Incentive Aware Routing based on Game Theory for Selfish Opportunistic Networks. *KSII Trans. Internet Inf. Syst.* **2019**, *13*, 152–171.

24. Xiong, Z.; Xiao, N.; Xu, F.; Zhang, X.; Xu, Q.; Zhang, K.; Ye, C. An equivalent exchange based data forwarding incentive scheme for socially aware networks. *J. Signal Process. Syst.* **2021**, *93*, 249–263. [[CrossRef](#)]
25. Xiong, Z.; Li, X.; Zhang, X.; Zhu, S.; Xu, F.; Zhao, X.; Wu, Y.; Zeng, M. A service pricing-based two-stage incentive algorithm for socially aware networks. *J. Signal Process. Syst.* **2022**, *94*, 1227–1242.
26. Zhang, X.; Rao, Y.; Xiong, Z.; Deng, H.; Xu, F.; Yuan, L. Resource-constrained and socially selfish-based incentive algorithm for socially aware networks. *J. Signal Process. Syst.* **2023**, *95*, 1439–1453.
27. Spyropoulos, T.; Psounis, K.; Raghavendra, C.S. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking, New York, NY, USA, 26 August 2005; pp. 252–259.
28. Cui, J.; Liu, S.; Chang, Y.; Liu, Q.; Wu, Q. Efficient Routing Algorithm Based on Virtual Currency Transaction in DTN. *Comput. Sci.* **2024**, *51*, 300–308.
29. Li, X.; Jiang, P.; Chen, T.; Luo, X.; Wen, Q. A survey on the security of blockchain systems. *Future Gener. Comput. Syst.* **2020**, *107*, 841–853. [[CrossRef](#)]
30. Deng, M.; Lyu, Y.; Yang, C.; Xu, F.; Ahmed, M.; Yang, N.; Xu, Z.; Ke, C. Lightweight Trust Management Scheme Based on Blockchain in Resource-Constrained Intelligent IoT Systems. *IEEE Internet Things J.* **2024**, *11*, 25706–25719. [[CrossRef](#)]
31. Matsumoto, S.; Reischuk, R.M. IKP: Turning a PKI Around with Decentralized Automated Incentives. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 410–426.
32. He, Y.; Li, H.; Cheng, X.; Liu, Y.; Yang, C.; Sun, L. A Blockchain Based Truthful Incentive Mechanism for Distributed P2P Applications. *IEEE Access* **2018**, *6*, 27324–27335. [[CrossRef](#)]
33. Zhang, W.; Wang, H.; Hua, H.; Wang, Q. Selfish Mining and Defending Strategies in the Bitcoin. *Intell. Autom. Soft Comput.* **2022**, *34*, 1861–1875. [[CrossRef](#)]
34. Keränen, A.; Ott, J.; Kärkkäinen, T. The ONE simulator for DTN protocol evaluation. In Proceedings of the 2nd International Conference on Simulation Tools and Techniques, Rome, Italy, 2–6 March 2009; ICST: Brussels, Belgium, 2010; pp. 1–10.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

FPGA Implementation of Pillar-Based Object Classification for Autonomous Mobile Robot

Chaewoon Park ¹, Seongjoo Lee ^{2,3} and Yunho Jung ^{1,4,*}¹ School of Electronics and Information Engineering, Korea Aerospace University, Goyang-si 10540, Republic of Korea; pcw0201@kau.kr² Department of Electrical Engineering, Sejong University, Seoul 05006, Republic of Korea; seongjoo@sejong.ac.kr³ Department of Convergence Engineering of Intelligent Drone, Sejong University, Seoul 05006, Republic of Korea⁴ Department of Smart Air Mobility, Korea Aerospace University, Goyang-si 10540, Republic of Korea

* Correspondence: yjung@kau.ac.kr; Tel.: +82-2-300-0133

Abstract: With the advancement in artificial intelligence technology, autonomous mobile robots have been utilized in various applications. In autonomous driving scenarios, object classification is essential for robot navigation. To perform this task, light detection and ranging (LiDAR) sensors, which can obtain depth and height information and have higher resolution than radio detection and ranging (radar) sensors, are preferred over camera sensors. The pillar-based method employs a pillar feature encoder (PFE) to encode 3D LiDAR point clouds into 2D images, enabling high-speed inference using 2D convolutional neural networks. Although the pillar-based method is employed to ensure real-time responsiveness of autonomous driving systems, research on accelerating the PFE is not actively being conducted, although the PFE consumes a significant amount of computation time within the system. Therefore, this paper proposes a PFE hardware accelerator and pillar-based object classification model for autonomous mobile robots. The proposed object classification model was trained and tested using 2971 datasets comprising eight classes, achieving a classification accuracy of 94.3%. The PFE hardware accelerator was implemented in a field-programmable gate array (FPGA) through a register-transfer level design, which achieved a 40 times speedup compared with the firmware for the ARM Cortex-A53 microprocessor unit; the object classification network was implemented in the FPGA using the FINN framework. By integrating the PFE and object classification network, we implemented a real-time pillar-based object classification acceleration system on an FPGA with a latency of 6.41 ms.

Keywords: field-programmable gate array; autonomous mobile robot; object classification; pillar feature encoder; FINN



Citation: Park, C.; Lee, S.; Jung, Y. FPGA Implementation of Pillar-Based Object Classification for Autonomous Mobile Robot. *Electronics* **2024**, *13*, 3035. <https://doi.org/10.3390/electronics13153035>

Academic Editor: Nicola Lusardi

Received: 14 June 2024

Revised: 29 July 2024

Accepted: 30 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Autonomous mobile robots move by recognizing their surroundings and performing physical tasks performed by people in the past. Advances in computer vision, robot control, and artificial intelligence (AI) have enhanced the ability of autonomous mobile robots to perceive and respond to their environment. Consequently, these robots are utilized in several areas of human life, including manufacturing, transportation, healthcare, and daily living [1,2]. Object classification is essential for autonomous mobile robots for identifying objects and planning travel paths [3].

Various types of sensors such as cameras and radio detection and ranging (radar) sensors have been used to perceive the surrounding environment in autonomous driving situations. Camera sensors cannot obtain depth information, making it impossible to accurately determine the distance to an object, which can lead to serious issues when the two objects overlap [4]. In addition, camera sensors are sensitive to brightness and have a limited

field of view, which results in operational constraints [3]. Radar sensors cannot determine the height information of objects and have very low resolution for angles, preventing the precise observation of objects [5]. In contrast, point clouds acquired by light detection and ranging (LiDAR) sensors contain 3D physical coordinate information that enables the acquisition of depth and height information regarding objects. Furthermore, there are fewer operational constraints, and the high resolution allows for precise observation of objects [3]. Therefore, the use of point clouds acquired by LiDAR sensors is preferred for autonomous driving.

Various deep learning algorithms are available for processing point clouds. MVCNN [6] projects a point cloud in multiple directions and classifies objects using a convolutional neural network (CNN) based on the projected images. Although it uses a simple method to convert 3D data into 2D images and utilize a CNN, information loss occurs during the projection process [7]. PointNet [8] is a permutation-invariant network designed using a symmetry function to handle unordered point sets. VoxelNet [9] encodes features on a voxel-by-voxel basis using PointNet and conducts inferences using a region proposal network. Because the inference is based on a 3D CNN, a long inference time of 225 ms is required [10]. SECOND [11] achieved an improved inference speed of 50 ms by applying a sparse CNN to VoxelNet, but the complexity of the 3D CNN computation remained high. PointPillars [12] proposed a pillar-based method for encoding 3D point clouds into 2D images. Unlike MVCNN, which only uses information from the highest element, the pillar-based method uses information from all points within the pillar. Conducting 2D CNN-based inference on an encoded image enables pillar-based methods to achieve high-speed inference and excellent performance [13–16].

Autonomous mobile robots must have real-time response time to prevent physical accidents. Under these constraints, running microprocessor unit (MPU)-based pillar-based methods are extremely time-consuming [17]. Graphic processing units (GPUs) can accelerate inference based on 2D CNNs; however, using GPUs on edge devices is inefficient in terms of power consumption [18]. Therefore, refs. [19–21] implemented a hardware accelerator for pillar-based methods using a field-programmable gate array (FPGA). However, a pillar feature encoder (PFE) was computed on an MPU in [19], requiring 71.2 ms and accounting for 19% of the total response time. Similarly, a PFE was computed on an MPU in 9.51 ms in [20], which was 22% of the total response time. Additionally, a portion of the PFE was accelerated in [21], requiring 37.7 ms for computation, which accounted for 59% of the total response time. Because the PFE occupies a significant portion of the overall response time of the pillar-based method, a hardware accelerator for PFE is useful.

The pillar-based method can perform inference on an image encoded by the PFE with a network based on a depth-wise separable CNN (DSCNN). The DSCNN used in [22–24] has been widely adopted in recent CNN models owing to its low computation cost and number of parameters. Because of these advantages, DSCNNs are often used in resource-constrained or latency-sensitive applications [25]. Additionally, despite their low complexity, they have shown successful performance in image classification models by enhancing the representational efficiency [26].

In this study, we propose an efficient pillar-based object classification model for autonomous mobile robots using a PFE and DSCNN, achieving a classification accuracy of 94.3% through training and validation with datasets acquired using Ouster OS1-32 [27]. In addition, we propose an FPGA-based acceleration system for the real-time operation of the proposed object classification model. The proposed acceleration system comprises a PFE accelerator and classification network accelerator. We designed the PFE accelerator to achieve a computation time of 0.23 ms and confirmed a 40 times acceleration performance compared with the MPU. A classification network accelerator was implemented on an FPGA using FINN. For operation in edge environments, we implemented the proposed acceleration system on an FPGA and confirmed a real-time response time of 6.41 ms.

The remainder of this paper is structured as follows. Section 2 provides an overview of the proposed acceleration system, including the PFE and DSCNN-based classification

networks, dataset used, model configuration, and evaluation results. The design of the proposed acceleration system is described in Section 3. Section 4 presents the implementation results and acceleration effects. Finally, Section 5 concludes the study.

2. Proposed System

Figure 1 shows an overview of the proposed acceleration system. The point cloud input to the system is encoded into an image by the PFE through three steps: (a) converting the point cloud into a set of pillar units, (b) adding hand-crafted features (HCFs), and (c) expanding the features through point-wise (PW) convolution, compressing them through a max-pool operation, and scattering them back to their original position on the x - y plane to generate an image. The generated image is then fed into a classification network to classify the objects. The classification network comprises a DSCNN-based network.

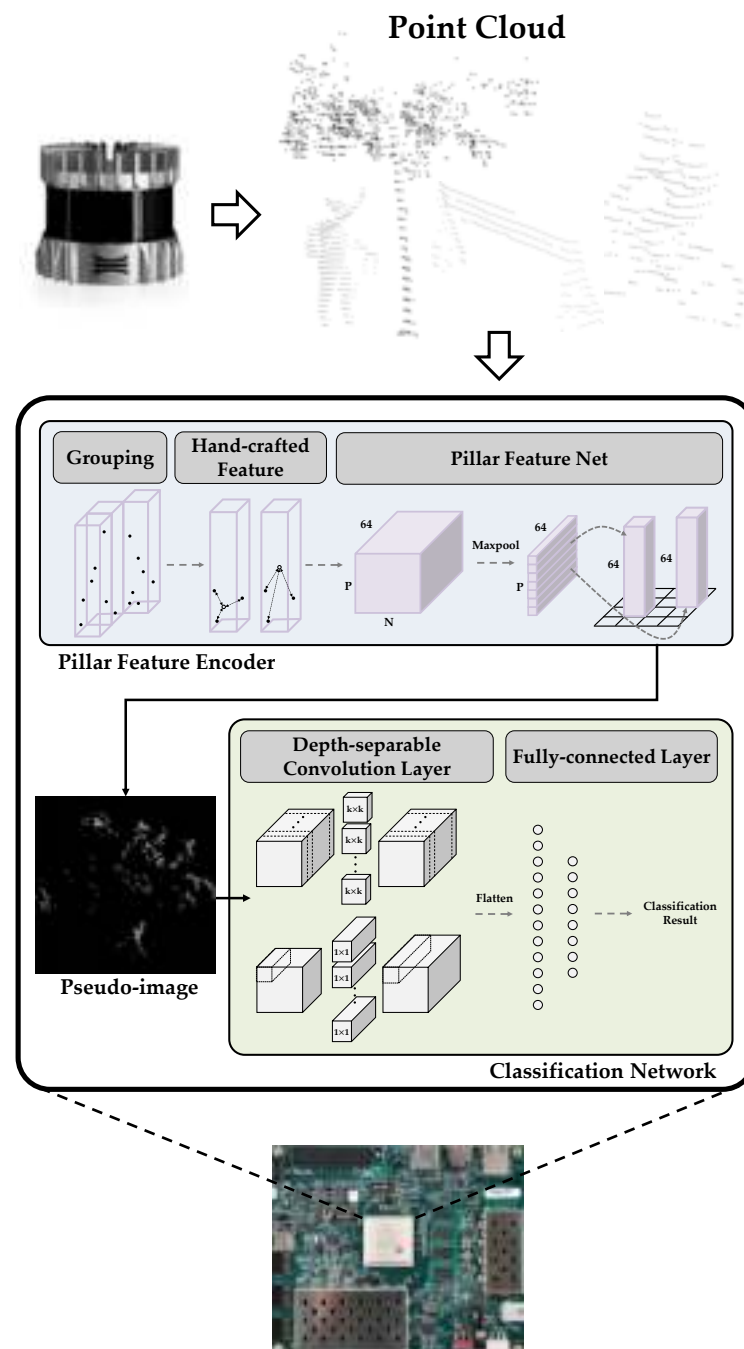


Figure 1. Overview of the proposed acceleration system.

2.1. PFE

The PFE, which encodes 3D point clouds into 2D images to enable 2D CNN-based inference, was proposed in [12]. First, a grid is generated by discretizing the x - y plane, and then the point cloud is converted into a set of pillar units. The point cloud set $\mathbf{g}$ is defined by Equation (1).

$$\mathbf{g} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_i, \dots, \mathbf{p}_P] \quad (1)$$

We set $\mathbf{g}$ as a set of $\mathbf{p}_i$, where P is the maximum number of pillars that contains points. We set $\mathbf{p}_i$ as defined by Equation (2).

$$\mathbf{p}_i = [\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_i, \dots, \mathbf{n}_N] \quad (2)$$

We set $\mathbf{p}_i$ as a set of $\mathbf{n}_j$, where N is the maximum number of points that a pillar can contain. The set $\mathbf{n}_j$ is defined as in Equation (3).

$$\mathbf{n}_i = [x, y, z, r] \quad (3)$$

The set $\mathbf{n}_j$ comprises the x , y , z coordinates and reflectivity values of the points. Finally, the point cloud is represented by a tensor of size $(P, N, 4)$, where P and N are hyperparameters. If the number of points in a pillar is greater than or equal to N , random sampling is used; if it is less than N , zero padding is used.

After the point cloud is converted into a set of pillar units, an HCF is added to set $\mathbf{n}_j$. The HCF comprises the differences x_m, y_m, z_m between the arithmetic mean coordinates of the points inside the pillar and x, y, z coordinates of each point, and the differences x_c, y_c between the center coordinates of the pillar and the x, y coordinates of each point. Setting $\mathbf{n}_j$ with the appended HCF is given by Equation (4).

$$\mathbf{n}_i = [x, y, z, r, x_m, y_m, z_m, x_c, y_c] \quad (4)$$

A tensor of size $(P, N, 9)$ with an added HCF undergoes PW convolution, batch normalization, and a ReLU layer to transform it into a tensor of size $(P, N, 64)$. Subsequently, maximum pooling is applied along the second dimension to yield a tensor of size $(P, 64)$. Finally, each pillar is scattered back to the original x - y plane, completing the 2D image data with 64 channels.

2.2. Classification Network

The proposed classification network is based on the DSCNN. Figure 2 shows how the DSCNN works. The DSCNN comprises the depth-wise (DW) and PW methods. In DW, the number of channels in the input image, filters, and channels in the output image are the same. Each filter has only one channel, and performs a convolution operation on only one channel in the input image to create a one channel output image. For example, in the DW convolution shown in Figure 2, one channel of the input image, separated by a dashed line, is performed a convolution operation with one filter to produce one channel of the output image, also separated by a dashed line. Let iC be the number of channels in the input image and oC be the number of channels in the output image; PW uses oC filters of size 1×1 with iC channels. Each filter performs a convolution operation on all the channels in the input image to create one channel of the output image. For example, in the PW convolution shown in Figure 2, one pixel of the input image, separated by a dashed line, completes one pixel of the output image, separated by a dashed line, by applying all filters in the convolution operation.

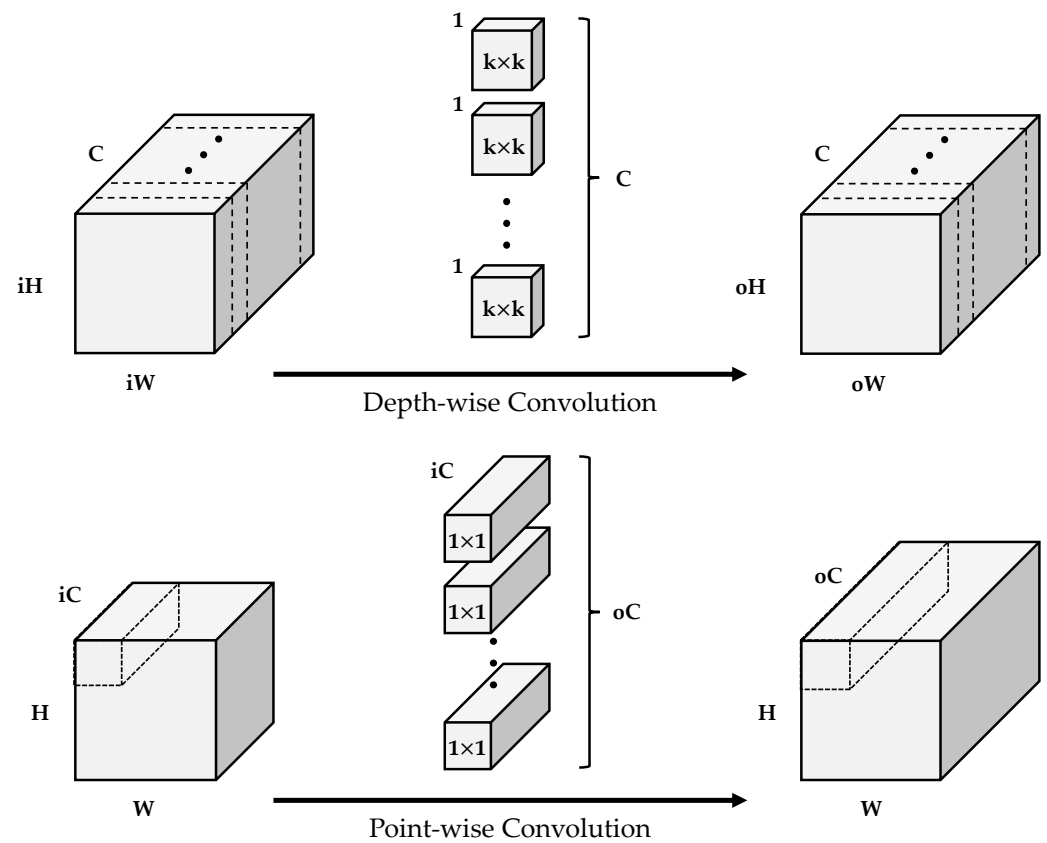


Figure 2. Operation mechanism of DSCNN.

2.3. Dataset

To train the proposed object classification model, we obtained point clouds using an Ouster OS1-32 sensor. Figure 3 illustrates examples from a dataset comprising objects typically encountered by autonomous mobile robots in their driving environments. To enhance the training quality, we augmented the dataset using samples from the nuScenes [28] dataset. The dataset includes classes such as building, tree, vehicle, bicycle, obstacle, greenery, person, and urban fixture (street light, bicycle rack, traffic light, etc.), as depicted in Figure 4, which shows the configuration of the dataset. Out of a total of 2971 data points, 2617 were used for training and 354 were used for validation.



Figure 3. Cont.

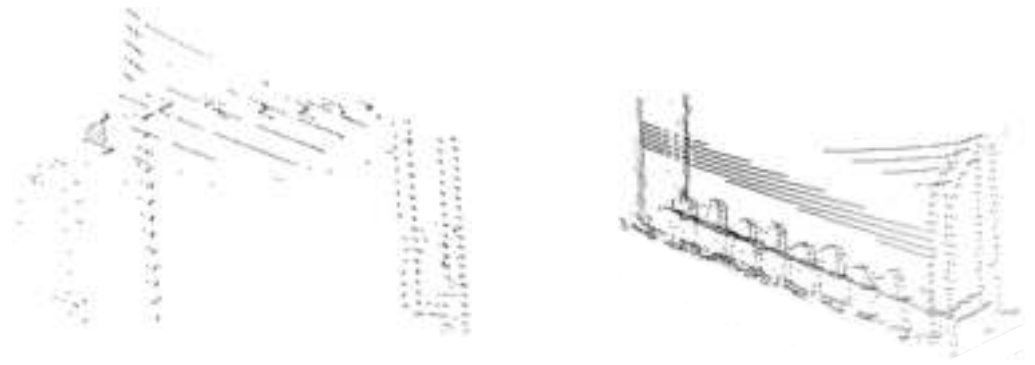


Figure 3. Examples of dataset.

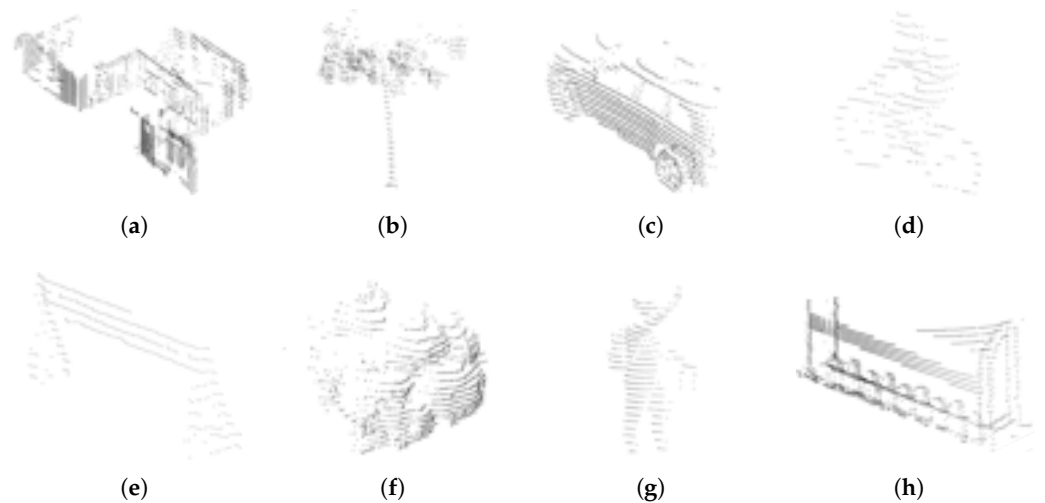


Figure 4. Configuration of dataset classes: (a) building; (b) tree; (c) vehicle; (d) bicycle; (e) obstacle; (f) greenery; (g) person; (h) urban fixture.

2.4. Performance Evaluation

In the PFE, the image size, number of pillars, and number of points per pillar are hyperparameters that significantly influence both the performance and complexity of the model. The image size determines the resolution and dimensions of the encoded image. A larger image size results in higher-resolution images; however, it also increases the computational demand of the CNN. Similarly, increasing the number of pillars and points per pillar allows more detailed information to be captured from the point cloud. However, this also increases the computational load of the PFE owing to the increased number of computations required to process a larger number of data. Thus, finding a balanced combination of these hyperparameters is crucial for achieving both high performance and efficient computation in pillar-based object classification models.

Figure 5 illustrates the architectures of the networks utilized for performance evaluation, all of which were based on the DSCNN. Table 1 presents the accuracies corresponding to various hyperparameters and network structures, with the overall accuracy serving as an evaluation metric for object classification. We trained 300 epochs with a batch size of 16, and the results of the best performing epochs are presented in Table 1. The training was performed with a learning rate of 0.001 and a weight decay rate of 0.0001. First, it was observed that Networks 1 and 5 exhibited lower performance than Networks 2, 3, and 4. Among Networks 2, 3, and 4, Network 4 stands out for its balanced performance, demonstrating comparable accuracy to Networks 2 and 3, while boasting lower computational complexity and fewer parameters. Additionally, the image size of 128×128 exhibited outstanding performance, whereas the performance was comparable when the number of pillars was 1024 and 512. The best performance was achieved when 16 points were inside

each pillar. Consequently, the proposed object classification model adopted an image size of 128×128 , 512 pillars, and 16 points within each pillar, with the classification network structured according to Network 4.

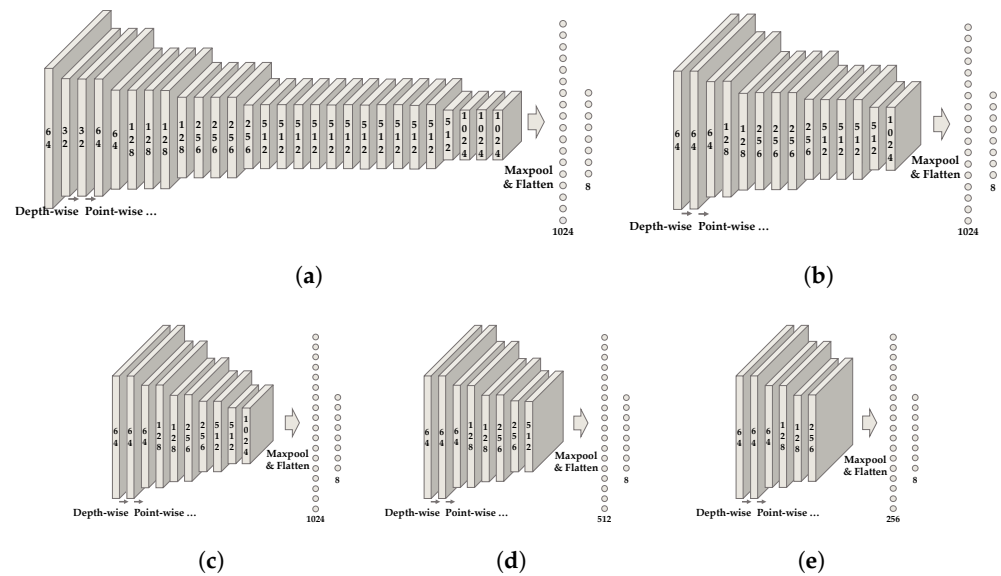
Table 1. Accuracy of model by various hyperparameters and networks.

Image Size	Hyperparameter		Network				
	Number of Pillars	Number of Points per Pillar	1	2	3	4	5
256×256	2048	32	91.8%	93.5%	93.5%	92.9%	89.7%
		16	92.9%	92.9%	92.4%	92.4%	88.0%
		8	92.9%	92.9%	92.9%	92.4%	89.7%
	1024	32	92.4%	92.4%	91.8%	92.9%	89.1%
		16	92.4%	92.9%	92.4%	92.9%	90.2%
		8	93.5%	93.5%	92.4%	92.4%	89.7%
	512	32	92.4%	92.9%	93.5%	93.5%	89.7%
		16	92.9%	94.0%	92.4%	93.5%	89.7%
		8	94.6%	92.9%	92.4%	91.8%	90.8%
128×128	1024	32	90.8%	91.8%	92.4%	94.6%	90.8%
		16	90.8%	94.6%	92.9%	92.4%	91.3%
		8	90.8%	91.8%	91.8%	94.0%	91.8%
	512	32	90.8%	94.0%	91.8%	93.5%	92.4%
		16	91.3%	92.9%	93.5%	94.6%	92.9%
		8	90.8%	93.5%	91.8%	92.9%	92.4%
	256	32	91.8%	93.5%	92.9%	91.8%	90.8%
		16	90.8%	92.9%	92.4%	91.3%	90.8%
		8	90.8%	94.0%	94.0%	92.4%	91.3%
64×64	512	32	89.7%	89.7%	92.9%	91.8%	91.3%
		16	88.6%	90.2%	93.5%	91.8%	91.8%
		8	88.6%	90.8%	92.9%	92.4%	92.4%
	256	32	88.6%	90.2%	93.5%	90.8%	90.8%
		16	87.5%	89.7%	93.5%	91.8%	91.8%
		8	87.5%	90.8%	93.5%	91.8%	90.8%
	128	32	88.6%	89.1%	89.7%	90.8%	89.7%
		16	89.1%	90.2%	91.3%	90.2%	90.8%
		8	86.4%	92.4%	91.3%	89.1%	91.8%

Table 2 presents a performance comparison between the proposed classification network and the other classification networks in this study. The proposed classification network achieved the highest classification accuracy of 94.6% based on the DSCNN, with a small number of multiply-accumulate (MAC) operations of 37 M and a small number of parameters of 50.8 K.

Table 2. Performance comparison with other networks.

Network	Accuracy	Number of MACs	Number of Parameters
LeNet	92.2%	787.6 M	2.0 M
ResNet18	83.8%	1.4 G	11.4 M
ResNet34	82.3%	2.0 G	21.5 M
MobileNet	85.7%	128.3 M	3.2 M
Ours	94.6%	37.0 M	50.8 K

**Figure 5.** Architecture of DSCNN-based object classification network: (a) Network 1; (b) Network 2; (c) Network 3; (d) Network 4; (e) Network 5.

3. Implementation of Acceleration System

The proposed acceleration system was implemented on an FPGA by integrating the PFE hardware accelerator intellectual property (IP) and classification network hardware accelerator IP with an MPU. Section 3.1 presents the bit width of each IP obtained through quantization experiments. Section 3.2 describes the structure and operations of the proposed acceleration system. Section 3.3 describes the structure and behavior of the PFE hardware accelerator, and Section 3.4 describes the FINN used to implement the object classification network hardware accelerator.

3.1. Quantization

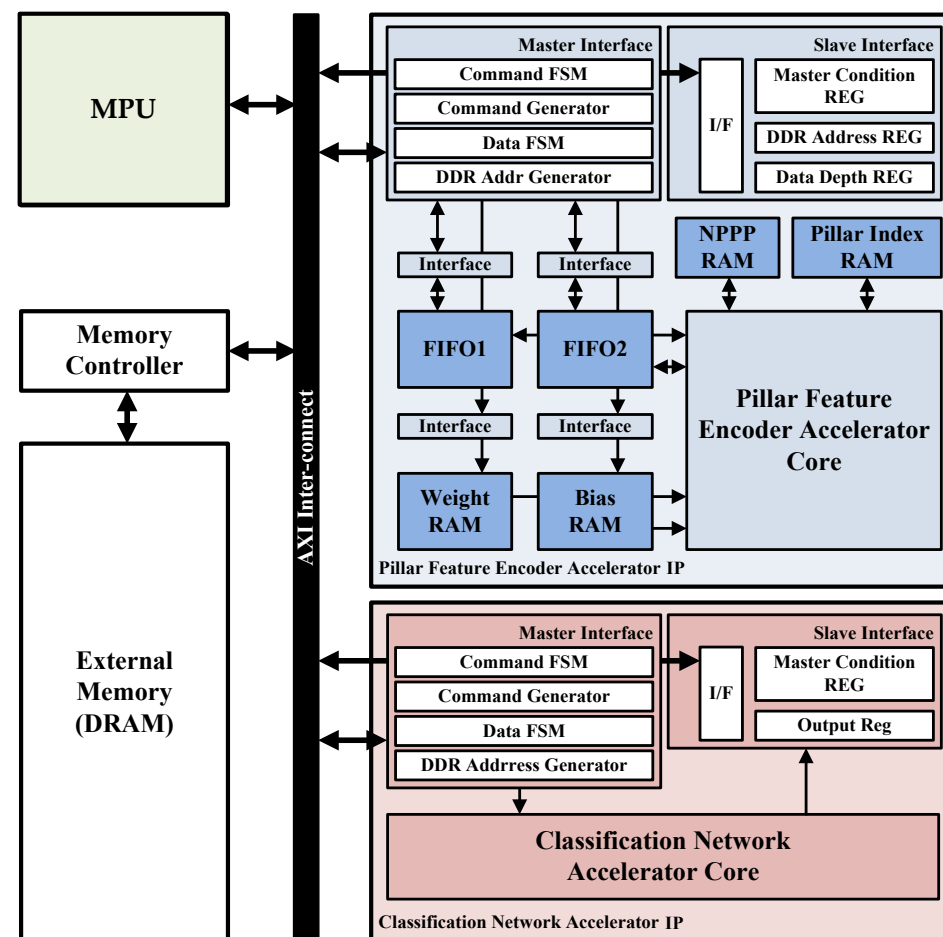
In this study, we quantized the proposed object classification model to implement the proposed acceleration system in the hardware. To minimize the performance reduction owing to errors caused by quantization, we performed quantization-aware training (QAT). The results of the experiments using different bit formats for the PFE and classification networks are listed in Table 3. The PFE is more accurate with 16 and 32 bits than with 8 bits, but there is no difference in accuracy between 16 and 32 bits. Therefore, we decided to use a 16 bit format for the PFE. The object classification network showed higher accuracy with 4 and 8 bits, but there was no difference in performance; therefore, we used 4 bits.

Table 3. Accuracy according to bit format of PFE and classification networks.

PFE	Classification Network		
	2	4	8
8	90.3%	91.7%	92.3%
16	89.7%	94.3%	94.3%
32	88.7%	92.7%	94.3%

3.2. System Architecture

Figure 6 shows the structure of the proposed acceleration system. To validate the system in an FPGA environment, the accelerators were integrated with an MPU via an advanced extensible interface (AXI) interconnect and implemented in an FPGA. Each accelerator IP is a slave to the MPU, and the MPU sends the start and end DRAM addresses that the IP will access, the number of words to be read from DRAM, AXI protocol parameters, and start and stop signals to the slave registers of the IP. The IP acts as a master to DRAM, generating read and write addresses through its master interface to fetch the required data from DRAM during computation and write back the results to DRAM.

**Figure 6.** Architecture of proposed acceleration system on FPGA.

The MPU first sets the slave register of the PFE accelerator IP, which starts by reading the point cloud from DRAM through the master interface to organize the point cloud into pillars. The PFE accelerator checks the coordinates of the point cloud and stores the point cloud in FIFO1 one after the other. To store the point cloud in DRAM pillar-by-pillar, it generates the DRAM addresses and stores them in FIFO2. When all the DRAM addresses

for the point cloud are generated and stored in FIFO2, the master interface references FIFO1 and FIFO2 to store the point cloud at an appropriate location in DRAM.

The PFE accelerator organizes the entire point cloud into pillars by appropriately storing it in DRAM, and then reads the point cloud back and stores it in FIFO1 and FIFO2. It also reads the weights and biases for performing PW convolution from DRAM and stores them in the weight memory (WM) and bias memory (BM). After adding the HCF to the point cloud, PW convolution is performed to expand the features, and a max-pool operation is performed. Thus, one pillar is encoded with 64 features, which is one pixel with 64 channels in a 2D image. The encoded pixel is stored in DRAM by considering its position.

After the PFE accelerator encodes all point clouds into 2D images, the MPU sets the slave register of the classification network accelerator IP and starts the accelerator to read images from DRAM through the master interface. After the classification network accelerator has read all images and has performed all computations, it writes the classification results to the slave register. The MPU completes the system operation by accessing the slave register of the classification network accelerator IP and reading the classification results.

3.3. PFE Hardware Architecture

Figure 7 shows a block diagram of the PFE accelerator. The proposed PFE accelerator comprises a grouping unit (GU), an HCF unit (HU), a PFN unit (PU), and eight memories. The memories are FIFO1, FIFO2, number of points per pillar (NPPP) buffer (NB), pillar index buffer (PB), HCF buffer1 (HB1), HB2, WM, and BM. HB1 and HB2 are utilized in a ping-pong scheme by the HU and PU within a pipelined structure. As the HU operates, it writes results to HB1, while the PU concurrently accesses data from HB2. Subsequently, the HU switches to writing computation results to HB2, while the PU utilizes the data stored in HB1 during this phase.

The point clouds are stored in FIFO1 in order, and the grid locator in the GU checks the coordinates of the points to obtain the coordinates of the pillars containing the points. For example, if the width of a grid cell is an integer value of 2^5 , the coordinates of the pillar containing a point can be determined by dividing the point's coordinates by 2^5 . This division operation is replaced by a 5 bit right shift operation. The coordinates of the pillars are used as the read address of the NB to determine the NPPP. If the NPPP is 0, a new pillar is created; thus, the pillar counter is incremented by one, and the value is written to the PB to be used as the index of the pillar. If the NPPP is less than 16, it is incremented by 1 and written back to the NB; otherwise, it remains unchanged. The address is generated by using the pillar index multiplied by 16 (the maximum NPPP) as an offset, and then adding the NPPP to this offset. The generated address is stored in FIFO2. With the point cloud in FIFO1 and the DRAM address in FIFO2, the point cloud is stored in the correct location in DRAM.

After all the point clouds are stored in DRAM in the pillars, they are read back into FIFO1 and FIFO2. In FIFO2, the DRAM write address is stored during the grouping operation, while the DRAM read data are stored at other times. The PFE accelerator then performs computation on all point clouds in a pillar-wise manner. The averaging unit (AU) of the HU adds the x , y , and z coordinates of all points in the pillar and is divided by the NPPP to obtain the arithmetic mean coordinates of the points. The offset unit in the HU computes x_m , y_m , z_m by subtracting the center coordinates of the points obtained from the AU from the x , y , z coordinates of each point, and x_c , y_c by subtracting the center coordinates of the pillar from the x , y coordinates of each point. The x , y , z , r , x_m , y_m , z_m , x_c , y_c of the points with the HCF added are then stored in the HB.

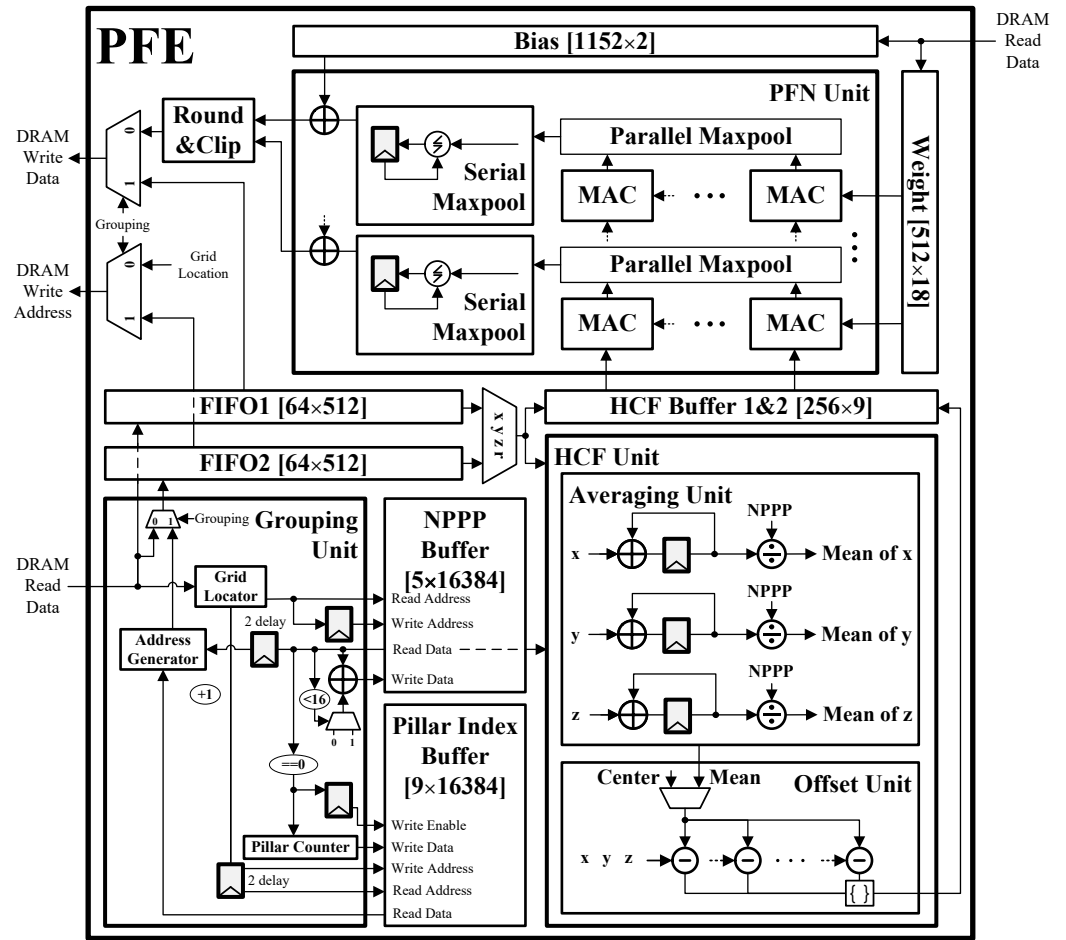


Figure 7. Block diagram of PFE accelerator.

Figure 8a shows the tensor-level operation performed by the PU. The PU executes PW convolution, batch normalization, max-pool, and ReLU operations on the HCF. The weights and biases of the batch normalization can be combined with those of the PW convolution layer. In the PW convolution, only the weights are used, and the biases are added after a biased ReLU operation, which thresholds values at $-bias$ instead of 0.

Figure 8b represents the operation in Figure 8a at the matrix level. The PU performs the multiplication of the HCF matrix of size (16, 9) and the weight matrix of size (9, 64) using an 8×32 MAC array. To do this, the HCF matrix is divided into two matrices of size (8, 9), and the weight matrix is divided into two matrices of size (9, 64), performing the entire matrix multiplication in four parts. For instance, the matrix multiplication of HCF tile 2 and weight tile 1 produces result tile 2, and these tiles are max-pooled to create max-pooled tile 1.

Figure 8c illustrates the hardware-level operation of the PU in chronological order. The PU reads HCF tile 1 and weight tile 1 from the HB and WM, respectively. The PU reads HCF tile 1 column by column and weight tile 1 row by row over 9 cycles, feeding them into the MAC array. The MAC array processes the HCF and weight inputs over 9 cycles and outputs result tile 1 of size (8, 32). Result tile 1 is then processed by the parallel max-pool unit to perform max-pool operations, producing intermediate tile 1 of size (1, 32). Because the parallel max-pool unit immediately performs max-pool operations on the MAC array output, it saves cycles and buffer memory space by not storing and reloading result tile 1. Intermediate tile 1 is compared to the $-bias$ value in the serial max-pool unit, and the larger value is stored in the register. By performing biased ReLU operations in the serial max-pool unit, the ReLU operation could be integrated into the max-pool operation. After creating intermediate tile 1, the PU reads HCF tile 2 and weight tile 1 to produce intermediate tile

2. Intermediate tile 2 undergoes max-pool operations with intermediate tile 1 stored in the serial max-pool unit's register, resulting in max-pooled tile 1. Max-pooled tile 1 is half a pixel when the bias is added and rounded and clipped. This half pixel is stored in the correct location in DRAM based on its coordinates in the image. To create the DRAM write address, the x coordinate of the pixel multiplied by the image's width is used as an offset, and the y coordinate is added to this offset. When the PU has created both max-pooled tiles 1 and 2, there is one complete pixel in DRAM.

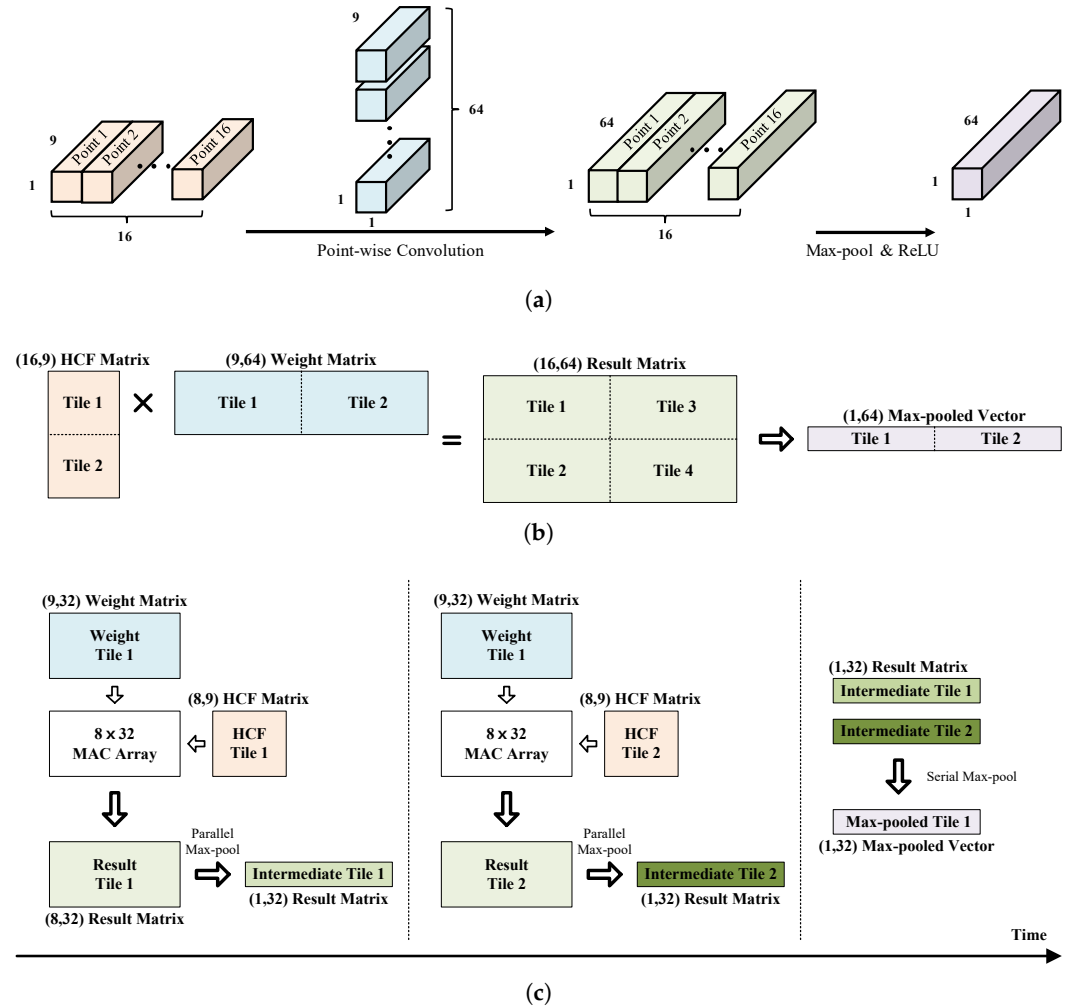


Figure 8. Operation of the PU: (a) tensor-level operation; (b) matrix-level operation; (c) hardware-level operation.

3.4. Classification Network Accelerator Using FINN

Xilinx offers various tools for implementing deep learning networks. Vitis AI is a platform used to implement and optimize artificial intelligence and machine learning applications on FPGAs and adaptive system-on-chip platforms. Vitis AI supports various artificial intelligence and machine learning frameworks and libraries, and generates a deep learning processing unit (DPU) IP. FINN is an open-source framework for FPGA-based quantized deep learning network inference. FINN uses an open neural network exchange model as the input, optimizes the network, and generates a hardware accelerator.

In this study, we used FINN to implement a hardware version of an object classification network. Whereas the DPU serves as a general-purpose accelerator for deep learning operations, FINN optimizes the hardware architecture and data flow specifically for the deep learning network. Consequently, FINN offers lower latency, enables real-time response speeds, and can achieve high throughput by optimizing the pipeline [29]. Moreover, net-

works with low complexity, such as the proposed classification network, can significantly reduce hardware resource usage. In addition, FINN supports 4 bit integers, providing an advantage over DPUs. Leveraging the benefits of FINN, ref. [19] implemented a backbone layer of PointPillars using FINN.

4. Implementation Results

The proposed acceleration system was implemented on a Xilinx Zynq Ultrascale+ ZCU104 [30]. Table 4 summarizes the hardware resource usage and operating frequency of the proposed acceleration system. The proposed acceleration system utilized 75,548 configurable logic block (CLB) look-up tables (LUTs), 56,931 CLB registers, 374 digital signal processors (DSPs), and 65 block RAMs. The operating frequency was 187.5 MHz, which enables high-speed operation, and the response time of the proposed object classification system was 6.41 ms.

Table 4. Hardware resources of proposed object classification system.

Unit	CLB LUTs	CLB Registers	DSPs	Block RAM	Frequency (MHz)
PFE	35,567	28,513	302	10.5	
Classification Network	34,788	21,625	72	54.5	
AXI Interconnect	5193	6793	0	0	
Total	75,548	56,931	374	65	187.5

Table 5 compares the response times of the firmware-based PFE implementation on an ARM Cortex A53 (baseline) operating at 1.2 GHz and the proposed PFE hardware accelerator (proposed) in terms of the execution time for each operation. The grouping operation, responsible for converting a point cloud into a set of pillar units, experienced a remarkable acceleration of 23.67 times, reducing from 0.71 ms to 0.03 ms. Notably, the HCF and PFN operations achieved a significant acceleration of approximately 42.6 times, decreasing from 8.52 ms to 0.20 ms. This notable acceleration was enabled by leveraging the parallel computation capability of the MAC array in the PFE accelerator and the pipelined structure of the HCF and PFN operations. The overall execution time for PFE operations was reduced from 9.23 ms to 0.23 ms, demonstrating a 40 times acceleration effect compared with the firmware-based implementation.

Table 5. PFE response time comparison with baseline.

Operation	Baseline (Firmware)	Proposed (Hardware)
Grouping	0.71 ms	0.03 ms
HCF	0.20 ms	
PFN	8.32 ms	0.20 ms
Total	9.23 ms	0.23 ms

Table 6 presents a comparison between the proposed object classification system and pillar-based methods implemented in previous studies. In [19–21], a heterogeneous system between the MPU and FPGA was constructed, whereas our system represents an end-to-end hardware implementation of all operations on an FPGA. In [19], PFE computation was performed on an MPU, and deep learning network computation was executed using an accelerator implemented through FINN. Ref. [20] computed the PFE on an MPU and executed a deep learning network on an accelerator implemented using a register-transfer level (RTL) design. Ref. [21] utilized both an MPU and DPU for PFE computation, and a DPU for deep learning network computation. We computed the PFE with an accelerator

implemented through an RTL design and deep learning network with an accelerator implemented through FINN.

Our method achieved the fastest response time of 6.41 ms, because it was the only one that accelerated the PFE, and the proposed object classification model was designed to have low computational complexity through various experiments. Moreover, these experiments resulted in an object classification model with fewer parameters. Our method used more DSPs than that in [19], but significantly fewer LUTs, registers, and memory. Ref. [19] used the same FPGA platform targeting edge environments as ours, the ZCU104, achieving low power consumption. However, it still consumed more power than ours, which implemented an efficient deep learning network. Compared with the method in [20], our method used fewer resources across the board, with particularly notable differences in DSPs and memory usage. Ref. [20] also consumed low power by implementing an efficient deep learning network. Additionally, since ref. [20] designed the deep learning network accelerator at the RTL level, it consumed even less power than ours, which was designed based on HLS. In addition, our method used more LUTs but fewer registers and DSPs, and substantially fewer BRAM and URAM than that in [21]. Furthermore, our method consumed significantly less power than that in [21]. Ref. [21] used the U280 FPGA platform targeting server environments, which had much higher power consumption and is not suitable for edge environments.

Table 6. Comparison with other pillar-based implementations.

	[19]	[20]	[21]	Ours
Platform	ZCU104	ZC706	U280	ZCU104
Computation Device	MPU & FPGA (FINN)	MPU & FPGA (RTL)	MPU & FPGA (DPU)	FPGA (RTL & FINN)
Execution Time (ms)	377.1	43.26	64.1	6.41
CLB LUTs	189,074	128,721	48,006	75,548
CLB Registers	150,187	111,118	85,801	56,931
DSPs	88	883	525	374
Block RAM	159	382.5	131.5	65
Ultra RAM	-	-	64	1
Frequency (MHz)	150	150	300	187.5
Power (W)	6.5	3.6	73.8	4.4

5. Conclusions

In this study, we propose a deep learning object classification model based on a pillar-based method for real-time responses of autonomous mobile robots. The object classification network of the deep learning model is based on the DSCNN, which has a low complexity and few parameters, making it suitable for edge device environments. The object classification model was trained using a dataset that considered the operating environment of autonomous mobile robots. To minimize the performance reduction due to quantization, we quantized and tested the model using QAT, achieving a classification accuracy of 94.3%. Because the computation of the PFE in pillar-based object classification models was time-consuming, we proposed a hardware accelerator for the PFE. The PFE accelerator was implemented through RTL design and achieved a response time of 0.23 ms, which was 40 times faster than the firmware program. The accelerator for the object classification network of the proposed deep learning model was implemented through the FINN framework and achieved an execution time of 6.18 ms. We also proposed an object classification system that integrates the proposed PFE accelerator and object classification network accelerator through an AXI interconnect. The proposed object classification system

was implemented and verified on an FPGA and achieved a real-time response time of 6.41 ms.

The proposed PFE accelerator can be combined with other deep learning network accelerators to accelerate more diverse deep learning models. However, in this study, we only performed acceleration on the object classification models. In future work, we plan to implement a system that supports acceleration for more diverse and challenging deep learning models.

Author Contributions: C.P. designed and implemented the proposed acceleration system, performed the experiment and evaluation, and wrote the paper. S.L. evaluated the proposed acceleration system and revised this manuscript. Y.J. conceived of and led the research, analyzed the experimental results, and wrote the paper. All authors read and agreed to the published version of the manuscript.

Funding: This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by Korean government (MSIT) (No. 2022-0-00960), and the CAD tools were supported by IDEC.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Varlamov, O. “Brains” for Robots: Application of the Mivar Expert Systems for Implementation of Autonomous Intelligent Robots. *Big Data Res.* **2021**, *25*, 100241. [\[CrossRef\]](#)
2. Liu, Y.; Li, Z.; Liu, H.; Kan, Z. Skill Transfer Learning for Autonomous Robots and Human–robot Cooperation: A Survey. *Robot. Auton. Syst.* **2020**, *128*, 103515. [\[CrossRef\]](#)
3. Yoshioka, M.; Suganuma, N.; Yoneda, K.; Aldibaja, M. Real-time Object Classification for Autonomous Vehicle using LIDAR. In Proceedings of the 2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Okinawa, Japan, 24–26 November 2017; pp. 210–211.
4. Gao, H.; Cheng, B.; Wang, J.; Li, K.; Zhao, J.; Li, D. Object Classification using CNN-based Fusion of Vision and LIDAR in Autonomous Vehicle Environment. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4224–4231. [\[CrossRef\]](#)
5. Zhou, Y.; Liu, L.; Zhao, H.; López-Benítez, M.; Yu, L.; Yue, Y. Towards Deep Radar Perception for Autonomous Driving: Datasets, Methods, and Challenges. *Sensors* **2022**, *22*, 4208. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-view Convolutional Neural Networks for 3D Shape Recognition. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 945–953.
7. Hoang, L.; Lee, S.H.; Lee, E.J.; Kwon, K.R. GSV-NET: A Multi-modal Deep Learning Network for 3D Point Cloud Classification. *Appl. Sci.* **2022**, *12*, 483. [\[CrossRef\]](#)
8. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
9. Zhou, Y.; Tuzel, O. Voxelnets: End-to-end Learning for Point Cloud based 3D Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4490–4499.
10. Bhanushali, D.; Relyea, R.; Manghi, K.; Vashist, A.; Hochgraf, C.; Ganguly, A.; Kwasinski, A.; Kuhl, M.E.; Ptucha, R. LiDAR-camera Fusion for 3D Object Detection. *Electron. Imaging* **2020**, *32*, 1–9. [\[CrossRef\]](#)
11. Yan, Y.; Mao, Y.; Li, B. Second: Sparsely Embedded Convolutional Detection. *Sensors* **2018**, *18*, 3337. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for Object Detection from Point Clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12697–12705.
13. Lis, K.; Kryjak, T. PointPillars Backbone Type Selection for Fast and Accurate LiDAR Object Detection. In Proceedings of the International Conference on Computer Vision and Graphics, Warsaw, Poland, 19–21 September 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 99–119.
14. Shu, X.; Zhang, L. Research on PointPillars Algorithm based on Feature-Enhanced Backbone Network. *Electronics* **2024**, *13*, 1233. [\[CrossRef\]](#)
15. Wang, Y.; Han, X.; Wei, X.; Luo, J. Instance Segmentation Frustum–PointPillars: A Lightweight Fusion Algorithm for Camera–LiDAR Perception in Autonomous Driving. *Mathematics* **2024**, *12*, 153. [\[CrossRef\]](#)
16. Agashe, P.; Lavanya, R. Object Detection using PointPillars with Modified DarkNet53 as Backbone. In Proceedings of the 2023 IEEE 20th India Council International Conference (INDICON), Hyderabad, India, 14–17 December 2023; pp. 114–119.

17. Choi, Y.; Kim, B.; Kim, S.W. Performance Analysis of PointPillars on CPU and GPU Platforms. In Proceedings of the 2021 36th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Jeju, Republic of Korea, 27–30 June 2021; pp. 1–4.
18. Silva, A.; Fernandes, D.; Névoa, R.; Monteiro, J.; Novais, P.; Girão, P.; Afonso, T.; Melo-Pinto, P. Resource-constrained onboard Inference of 3D Object Detection and Localisation in Point Clouds Targeting Self-driving Applications. *Sensors* **2021**, *21*, 7933. [CrossRef] [PubMed]
19. Stanisiz, J.; Lis, K.; Gorgon, M. Implementation of the Pointpillars Network for 3D Object Detection in Reprogrammable Heterogeneous Devices using FINN. *J. Signal Process. Syst.* **2022**, *94*, 659–674. [CrossRef]
20. Li, Y.; Zhang, Y.; Lai, R. TinyPillarNet: Tiny Pillar-based Network for 3D Point Cloud Object Detection at Edge. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *34*, 1772–1785. [CrossRef]
21. Latotzke, C.; Kloeker, A.; Schoening, S.; Kemper, F.; Slimi, M.; Eckstein, L.; Gemmeke, T. FPGA-based Acceleration of Lidar Point Cloud Processing and Detection on the Edge. In Proceedings of the 2023 IEEE Intelligent Vehicles Symposium (IV), Anchorage, AK, USA, 4–7 June 2023; pp. 1–8.
22. Tan, M.; Le, Q. Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
23. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
24. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
25. Lu, G.; Zhang, W.; Wang, Z. Optimizing Depthwise Separable Convolution Operations on GPUs. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *33*, 70–87. [CrossRef]
26. Kaiser, L.; Gomez, A.N.; Chollet, F. Depthwise Separable Convolutions for Neural Machine Translation. *arXiv* **2017**, arXiv:1706.03059.
27. Ouster. Ouster OS1 Lidar Sensor. Available online: <https://ouster.com/products/hardware/os1-lidar-sensor> (accessed on 22 May 2024).
28. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuscenes: A multimodal Dataset for Autonomous Driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11621–11631.
29. Xilinx Inc. Vitis™ AI Documentation Frequently Asked Questions. Available online: <https://xilinx.github.io/Vitis-AI/3.0/html/docs/reference/faq.html#what-is-the-difference-between-the-vitis-ai-integrated-development-environment-and-the-finn-workflow> (accessed on 22 May 2024).
30. AMD. UltraSclae+ ZCU104. Available online: <https://www.xilinx.com/products/boards-and-kits/zcu104.html> (accessed on 22 May 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Swin-Fake: A Consistency Learning Transformer-Based Deepfake Video Detector

Liang Yu Gong ^{*,†,‡} , Xue Jun Li ^{*,†,‡}  and Peter Han Joo Chong [†] 

Department of Electrical and Electronic Engineering, Auckland University of Technology,
Auckland 1010, New Zealand; peter.chong@aut.ac.nz

* Correspondence: liangyu.gong@aut.ac.nz (L.Y.G.); xuejun.li@aut.ac.nz (X.J.L.)

† Current address: 6 Saint Paul Street, Auckland 1010, New Zealand.

‡ These authors contributed equally to this work.

Abstract: Deepfake has become an emerging technology affecting cyber-security with its illegal applications in recent years. Most deepfake detectors utilize CNN-based models such as the Xception Network to distinguish real or fake media; however, their performance on cross-datasets is not ideal because they suffer from over-fitting in the current stage. Therefore, this paper proposed a spatial consistency learning method to relieve this issue in three aspects. Firstly, we increased the selections of data augmentation methods to 5, which is more than our previous study's data augmentation methods. Specifically, we captured several equal video frames of one video and randomly selected five different data augmentations to obtain different data views to enrich the input variety. Secondly, we chose Swin Transformer as the feature extractor instead of a CNN-based backbone, which means that our approach did not utilize it for downstream tasks, and could encode these data using an end-to-end Swin Transformer, aiming to learn the correlation between different image patches. Finally, this was combined with consistency learning in our study, and consistency learning was able to determine more data relationships than supervised classification. We explored the consistency of video frames' features by calculating their cosine distance and applied traditional cross-entropy loss to regulate this classification loss. Extensive in-dataset and cross-dataset experiments demonstrated that Swin-Fake could produce relatively good results on some open-source deepfake datasets, including FaceForensics++, DFDC, Celeb-DF and FaceShifter. By comparing our model with several benchmark models, our approach shows relatively strong robustness in detecting deepfake media.

Keywords: consistency learning; Swin Transformer; consistency loss; deepfake detection



Citation: Gong, L.Y.; Li, X.J.; Chong, P.H.J. Swin-Fake: A Consistency Learning Transformer-Based Deepfake Video Detector. *Electronics* **2024**, *13*, 3045. <https://doi.org/10.3390/electronics13153045>

Academic Editors: Petia Georgieva,
Krasimir Tonchev and Agata
Manolova

Received: 1 July 2024
Revised: 26 July 2024
Accepted: 29 July 2024
Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Deepfake generation and detection both kept developing in an attacker–defender game from 2019 to 2021; however, with the rapid development of GAN-based methods [1,2] and Diffusion Models (DMs) [3], the powerful generative abilities of these models with advanced post-processing, such as color blending, make synthetic faces look more realistic. Thus, it is harder to distinguish one single video frame's authenticity with human eyes in a deepfake video nowadays.

Even though convolutional neural network (CNN)-based detectors [4–6] have had a remarkable effect on deepfake classification work, they still suffer from large reliance on training data, which results in the accuracy or Area Under the Curve (AUC) dropping considerably in cross-dataset experiments. Thus, in recent years, deepfake detectors tend to be designed as consistency learning methods to explore the spatial or temporal consistency, self-consistency or inconsistency between representations or patches. Additionally, Vision Transformers [7,8] have been proven to have strong applications, especially in image classification. Transformer-based deepfake detectors are becoming mainstream in this field because of their unique attention mechanisms that can be applied to generate spatial and

temporal features. Compared with other Vision Transformers [9,10], Swin Transformer can considerably enhance classification performance by setting window partition, cyclic shift and relative position bias mechanisms.

In this paper, we proposed a simple consistency learning framework utilizing Swin Transformer [11] as a feature extractor to explicitly extract and learn the consistency between captured video frames. Compared with traditional deepfake detectors, our approach also contains a data pre-processing block and a backbone model for feature extraction, but our loss combines the frames' consistency and classification loss. As for the data pre-processing work, we divided one video into several frames as a group to classify deepfake videos, which can be regarded as an ensemble technique. Unlike other ensemble learning techniques, we enriched feature extraction to help the model learn different forgery features with random and different data augmentation methods. In addition, traditional deepfake detectors always utilize some unchanged data augmentation methods such as basic transformation and normalization; however, for our data pre-processing block, we chose multiple advanced data augmentation methods to increase the randomness of data enhancement and prove that no matter how much the data are enhanced, the same category of data can be maintained with good consistency. As shown in Figure 1, each video was equally divided into frames to ensure the facial area is evenly distributed across the duration of the video. Our data pre-processing block also contains multiple data augmentations and is randomly applied on each captured frame to enrich different inputs. Then, a shared Transformer-based encoder extracts representations and calculates the similarity of pairs by making combinations to form consistency loss. Finally, a supervised classifier assigns the label for each extracted frame feature.

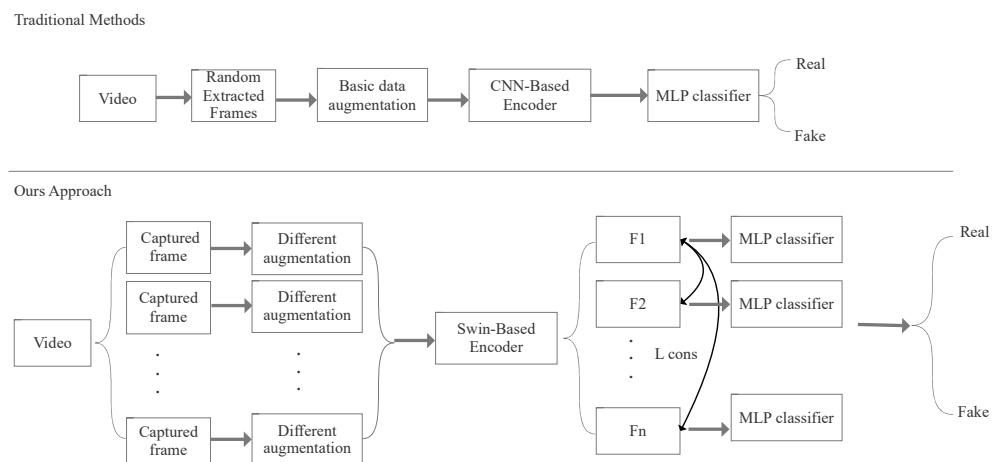


Figure 1. Illustration of Swin-Fake framework compared with traditional deepfake video detector framework.

There are three main improvements in our method compared with traditional deepfake detectors: (1) We utilize five data augmentation methods, including base transformation, Random Erasing [12], Random Augmentation, Random Crop and DFDC Selim [13], which are randomly applied, and they will largely avoid over-fitting at the input stage. (2) Swin Transformer is chosen as the encoder, which will combine the correlations between different divided image patches. (3) The final total loss is determined by two components, which are consistency loss and cross-entropy loss, and the decision is guaranteed by consistency learning.

2. Related Work

2.1. Deepfake Detection

Deepfake detection is always defined as a binary classification or a multi-class classification work because it essentially requires the model to distinguish the media as real or fake. The most popular framework is to utilize vanilla Xception [14] or ResNet [15] as backbones

to encode and then combine contrastive learning or explore representations' consistency. Capsule Network [4] is an effective lightweight network which can solve "Inverse Graphic" problems and proposes a dynamic routing algorithm to regroup the extracted features. CORE [16] is a consistency learning method based on Xception Network which improves the number of data augmentations to prove the model's consistency. Youtu-Lab [17] proposed a dual-contrastive learning architecture to distinguish the authenticity of faces using different data augmentations as well. Ensemble learning deepfake detectors could also significantly improve classification performance and have been widely applied in recent research; for example, Zhang [18] proposed a heterogeneous feature ensemble learning method to first extract gray gradient features, spectrum features and texture features from images, and then integrate them into an ensemble feature vector through a flattening process. Using DeepfakeStack, the authors of [19] chose to combine several state-of-the-art deepfake classification models to create an improved composite classifier which could reach 99.65% accuracy in some deepfake datasets. However, our proposed method, Swin-Fake, aims to explore the ability of Swin Transformer to extract forged features, such as PRNU noise and boundary mismatch, and then extract different frames of one video to prove the model's consistency in the classification process using different data augmentations.

2.2. Swin Transformer

Swin Transformer [11] is a hierarchical Vision Transformer that improves multi-head attention blocks (MSAs) to window multi-head attention (W-MSAs) and shifted window multi-head attention blocks (SW-MSAs). It layers different sizes of features by setting up different sizes of windows, which is different from the traditional Vision Transformer [7], and could save more on computing costs than the traditional Vision Transformer. Normally the input image has a size of $3 \times 3 \times 224$ and the output feature size is $7 \times 7 \times 768$. This framework contains one patch portion layer, one linear embedding layer and twelve Swin Transformer blocks to extract features, and the features are correlated with different patch embeddings. This network structure can more effectively simulate the sense of consistency and inconsistency of different patches when human eyes are observing and scanning different deepfake facial areas. Moreover, we investigated existing papers related to Transformer-based deepfake detectors, and surprisingly found there are not many papers focused on spatial consistency learning with a Transformer backbone. Most of them put more effort into designing temporal feature extraction blocks to obtain more temporal consistency information. Recently, Ilyas et al. [20] proposed an AVNet to detect audio-visual media using Swin Transformer, and Khalid [21] proposed utilizing Swin Y-Net for deepfake detection.

2.3. Data Augmentation

In the computer vision field, it is important to apply data augmentation, which is the first stage in preventing overfitting and an effective method to increase the amount of data. In this paper, we only utilize five image manipulation methods to change pixel values and positions. Base transformation is employed to combine pixel normalization and resizing. Random Erasing [12] is a cutout data transformation that randomly selects a rectangle block of the facial area and changes its color to totally black. Random Crop is used to crop the facial area in a random ratio and then resize it to a specific height and width. DFDC Selim is a complex data augmentation containing Gaussian noise, Gaussian blur, random shift and scale, and has already been proven to be the most useful data augmentation in deepfake classification work [16]. Random Augmentation randomly selects either Random Erasing, Random Horizontal Flip or Random Resize Crop. Compared with the design of traditional deepfake detectors, we have dramatically improved the diversity and randomness of input data, and it can be used for the subsequent verification of the similarity between different video frames of one video clip.

3. Proposed Methods

In this work, we outline the deepfake image detector pipeline and divide it into four parts: data pre-processing (including facial detection, crop and data augmentations), feature extraction, consistency calculation and prediction through classifiers. As shown in Figure 2, we first equally capture the frames of one detected video and randomly select and apply data augmentations to create different data views. After extracting the features through Swin Transformer, we separately calculate the cosine distances between representations in combinations and subtract the mean cosine distance as a consistency loss (See Algorithm 1). To prevent the unbalanced classes problem, we combine other databases' real videos into our training set, which is very important for further evaluation; in addition, we also choose to up-sample the real video frames; for example, if every fake video is equally divided into eight frames, the real video frame is equally divided into 16 or 32 correspondingly. This could largely prevent the trained model's predicted output from remaining in the same class.

Algorithm 1 The workflow of Swin-Fake

Input: The pretrained Swin Transformer: $f(x)$; video sequences: X
 Equally divide each video into M frames.
 Facial Crop and expand width and height to obtain facial images.
 Randomly select data augmentation methods to avoid overfitting.
for $i = 1$ to M **do**
 Utilize input sequences X to group set.
 Calculate the attention score between different divided patch embeddings.
 Compute $f(x)$ via forward propagation to obtain 2-D features F_x .
 Calculate cosine distance L_{Cos} between C_M^2 pairs using Equation (3).
 $L_{Cons} = \frac{1}{M} \sum_1^M mean(L_{Cos})$
 Utilize linear layer to project F_x to 2 classes.
 $L_{CE} = -\sum_1^N [y \log(p) + (1 - y) \log(1 - p)]$
 $L_{tot} = \alpha L_{Cons} + (1 - \alpha) L_{CE}$
 Conduct backward propagation and save the gradients for each layer.
end for
Output: a trained Swin-Fake model $f(x)$.

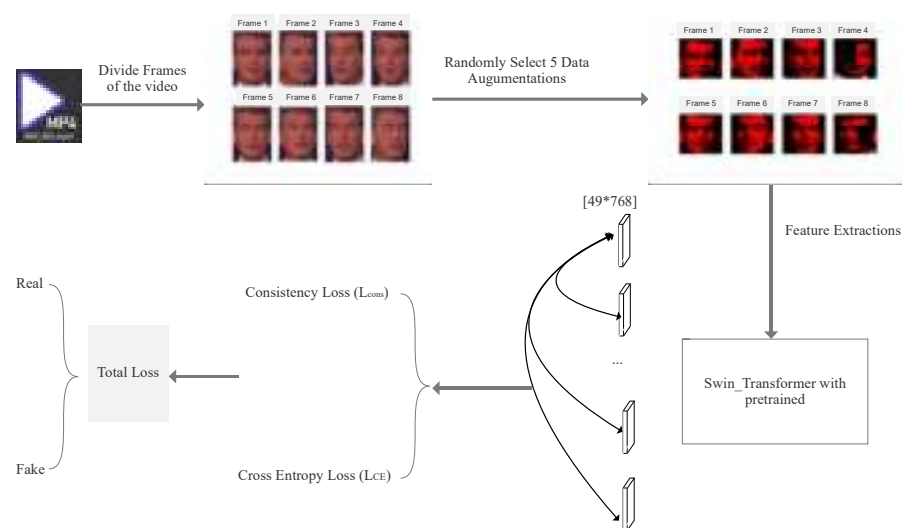


Figure 2. Explanation of Swin-Fake architecture. We firstly divide M frames of one video and apply different data augmentations for each frame. Then, they are sent to a shared Swin Transformer encoder. After obtaining M representations, they will be sent to separately calculate classification loss and consistency loss.

3.1. Pre-Processing work

Firstly, we give an input batch of B videos and equally divide each video into M video frames based on calculating each video's total frame number. The pre-processing stage also embeds a facial detector, "SCRFD" [22] to locate facial areas, align the faces and crop them to create specific datasets. Specifically, this facial detector is based on Insightface and utilizes specific Sample Redistribution (SR) and Computation Redistribution (CR) methods for facial detection. "scrfd 10g kps.onnx" is one of the model weight files utilizing this face detection method [22], and kps represents the 5 facial key points detected in this model. To preserve some background information, we expand the width and height of the cropped images by 10%. Then, the total $M \times B$ frames are neatly arranged in order, serving as the inputs for the Swin Transformer, and all resized to $3 \times 224 \times 224$. Since we apply and randomly selected five data augmentations in a transformation set T on $M \times B$ frames, the input vector sets I have a size of $[B, M, 3, 224, 224]$ which means every group contains $M [3, 224, 224]$ views. Finally, the input view set is written as:

$$I = \{\{x_1, x_2 \dots x_M\}_1 \dots \{x_1, x_2 \dots x_M\}_B\} \quad (1)$$

where x_n represents each frame view, B represents a batch of video, $\{x_1, x_2 \dots x_M\}_B$ represents the B -th group of data views and M represents how many frames are equally divided.

Once the input data views are obtained, five data augmentation methods will be randomly applied on these data views. These augmented tensors are the final input of the Swin-Fake model's inputs, as shown in Figure 3.



Figure 3. Examples of random augmented data. For Random Erasing, we set the Random Erasing that will be performed to 0.8, the scale is from 0.02 to 0.2 and the erased area ratio is from 0.5 to 2.0. For Random Crop, we set the Random Crop scale from 0.77 to 1 and Random Crop ratio from 0.9 to 1.1.

3.2. Encoder Network

We adopt Swin Transformer [11] as our feature extractor f because of its unique hierarchical structure. It contains multiple stacking Transformer blocks, among which the lower-level blocks can capture some color and edge information and the higher-level blocks can be utilized to analyze the semantic information. After applying the shifted window partitioning approach and linear embedding, we can change the dimension of the input feature and obtain the query (Q), key (K) and value (V) vectors separately. The queries (Q) and keys (K) have the same dimension of d . By computing the dot products of the query (Q) with all keys (K) and dividing each by $\sqrt{d}$, the first step of the attention correlation is finished. Then, we add the bias and apply a softmax function to squeeze the output within the range of 0 to 1 and obtain a value score. This attention calculation method aims to compute the weighted sum of values, which is the same as "Scaled Dot-product Attention". The calculation of self-attention is written as

$$Attention(Q, K, V) = SoftMax(QK^T / \sqrt{d} + B)V \quad (2)$$

where $1/\sqrt{d}$ is the scaling factor of Q, K and B is the bias of Q, K, V .

When the input set I goes through the encoder and is mapped into $B \times M$ representations, the representation $f(x_n)$ is a two d-dimensional vector with a size of $[49, 768]$.

Meanwhile, each M representation is set as one group and fed into the computation of consistency loss. After applying a linear layer on the last basic Swin block, we send the representations to compute Cross Entropy loss for classification. Thus, each batch of frames requires going through multiple data-augmented operations, and then, we can obtain the extracted low-dimensional features (See Figure 4). Finally, there will be two output vectors calculated, one for consistency loss calculation and another operated by the linear layer for classification loss. A detailed explanation of these two loss function calculations can be seen in Section 3.3.

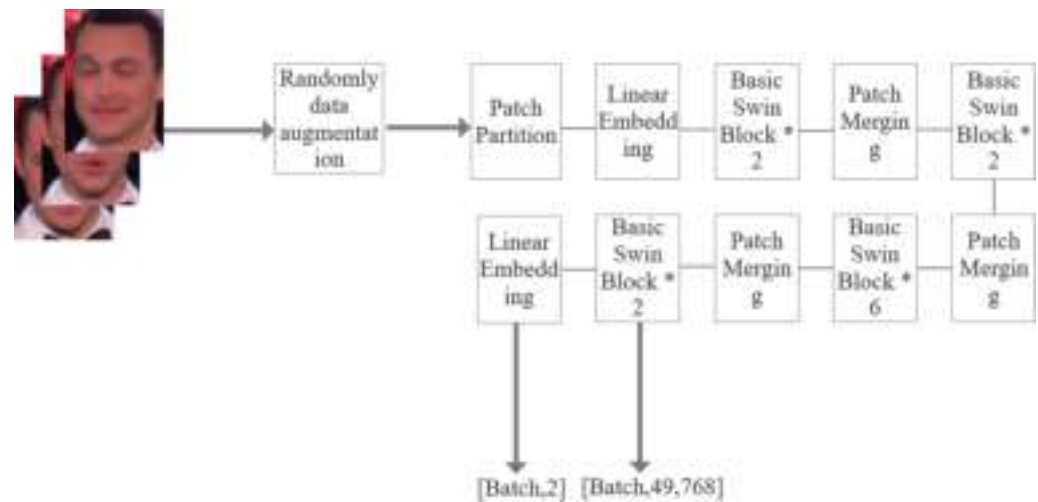


Figure 4. The architecture of the Swin Transformer backbone. * represents how many Basic Swin Blocks are applied in the backbone. There are two output vectors for each input data view set in our model, which are the extracted low-dimensional features with a size of $[M \times 49 \times 768]$ and the predicted probability $[M \times 2]$.

3.3. Loss Functions

The loss function of this model is designed with two components: classification loss and consistency loss. Consistency loss is applied on each group of M extracted representations and we subtract their distances. In addition, the reason we adopt the cosine distance to complete the consistency loss design is that we do not force each pair of representations to be exactly the same but to be more and more similar. The cosine distance calculation of each pair of representations can be written as

$$L_{Cos} = (1 - f_{x_i} \cdot f_{x_t}) \quad (3)$$

where x_i and x_t represent the i -th and t -th frames of data views, and $f_{(x_i)}$ and $f_{(x_t)}$ represent the normalized representations extracted by a shared encoder.

Since we have obtained augmented video clips with the amount of B groups as the input of our model, we also need to calculate the similarity between these features of the captured video clips. The aim of this is to solve some incorrectly distinguished results of hard samples by shortening their cosine distances, and we regard this as consistency loss. As for another factor that influences the total loss, consistency loss is designed to calculate the mean cosine distances of B batch videos. Specifically, the extracted M representations of one video can generate C_M^2 pairs of similarities through the above equation. The single consistency of one video can be expressed as the mean of C_M^2 pair similarity, and the total consistency loss of B batch videos requires us to calculate the average value of all the calculated single consistencies. Therefore, the consistency loss can be written as

$$L_{Cons} = \frac{1}{B} \sum_1^B \text{mean}(L_{Cos}) = \frac{1}{B} \sum_1^B \text{mean}(1 - f_{x_i} \cdot f_{x_t}) \quad (4)$$

where B represents the batch size of the video.

The main purpose of setting up consistency loss is to prove that no matter how realistic one particular deepfake video frame is, the model is able to shorten its spatial distance with other forgery frames. On the other hand, consistency loss validates the classification consistency of different frames of one video. Since the classification work is inseparable from supervised labels, we apply standard cross-entropy loss as classification loss, and we formulate it as follows:

$$L_{CE} = - \sum_1^{M \times B} [y \log(p) + (1 - y) \log(1 - p)] \quad (5)$$

where y is the ground truth label, p is the predicted probability, and $M \times B$ represents the total number of video frames.

Because our model is a consistency learning model, we utilize two loss functions (consistency loss and cross-entropy loss) in the linear combination to form our total loss. However, we set the hyper-parameter α as the weight balance to observe what proportion of these two loss functions can reach the minimum total loss and reach the highest accuracy. In Section 4.4, we conduct one ablation test to observe what value of this hyper-parameter can predict the most samples by traversing the weight parameter by adding 0.1 in every experiment, because we hope to find out the best fitting weight to balance these two loss functions, and we do not hope the total loss is largely biased toward one of the loss functions' results.

$$L_{tot} = \alpha L_{Cons} + (1 - \alpha) L_{CE} \quad (6)$$

where α is the hyper-parameter to balance cross entropy loss and consistency loss, and it ranges from 0 to 1.

4. Experiments

4.1. Datasets

To evaluate our method, we selected four common datasets in this field: FaceForensics++ [23], DFDC [24], Celeb-DF [25] and FaceShifter [26]. In particular, the first two datasets were used for the training phase, and Celeb-DF V2 and FaceShifter were used for cross-dataset experiments to prove our model's generalization ability. FaceForensics++ is a pioneering large-scale dataset in facial manipulation which contains 1000 real videos and 4000 fake videos generated by Face2Face, FaceSwap, Neural Textures and Deepfakes. DFDC is the largest datasets of these four datasets, containing 119,197 videos filmed by real actors. Compared with the previous two datasets, Celeb-DF is derived from 590 original YouTube videos and 5639 manipulated videos with advanced color blending, which is hard to distinguish by human eyes. FaceShifter is one sub-class of FaceForensics++ datasets, and it contains 1000 fake videos with different shooting scenarios. Considering the imbalanced training data of FaceForensics++, we randomly combined 2000 real videos from the DFDC database with real FF++ videos to solve this problem. As for the DFDC training, we chose to down-sample some fake video frames to solve the data imbalance.

4.2. Implementation Details

For data pre-processing, we extracted M frames of each video by setting a confidence threshold of 0.5 and an NMS threshold of 0.5 for "scrfd 10g kps.onnx". In addition, the detected cropped faces with bounding boxes were enlarged by 1.2 both in width and height to ensure there existed some background information on the model's inputs. As for the data augmentation settings, we set an 80% probability of erasing the facial area, the range of the proportion of the erased area against the input image was set from 0.02 to 0.2, and the range of the aspect ratio of the erased area was set from 0.5 to 2.0 in the Random Erasing operation. In the Random Crop augmentation, we set the scale's lower and higher bounds to 0.77 and 1, which means the random area of cropping before resizing was from 0.77 to 1 correspondingly. During the experiment, some selected frames (See Figure 5) with

no human faces were copied from the last detected index frame instead of being directly sent to the input data view set.



Figure 5. Some deepfake samples processed by “scrfd 10g kps.onnx”, which is the original input for our model training.

For model training, we set the video mini-batch size to 4 with an image size of 224×224 , and then utilized an AdamW optimizer [27] with weight decay set to 0.05 and the learning rate set to 0.0001. To effectively train this model, we stopped training early if there was no increase in validating accuracy in the first 5 epochs, and the total epoch was set to 30. In addition, all the dropout rates were set to 0.2 avoid over-fitting in our experiment. All our models used swin-tiny-patch4-window7-224.pth and were pre-trained on ImageNet-1K as the backbone, and the output features were extracted by the linear layer with a size of $[M, 49, 768]$ and the vectors with a size of $[M, 2]$. We initially adopted a balanced weight of 0.5 because we regarded consistency loss as having the same importance as classification loss at the beginning.

4.3. In-Dataset Study

In this section, we evaluate our proposed model’s validating performance in the same sources as the training datasets. For the in-dataset experiments, we assessed DFDC and FaceForensics++, which are two relatively large datasets, because we found that Swin Transformer always requires a large amount of training data to reach a good validation level during training experiments; otherwise, it will present a relatively low starting validating accuracy in some small-scale training datasets. And the initial hyper-parameter was set to 0.5 as default. We compared our consistency learning method with three baselines: Capsule Network, STIL [28] and ISTVT [10]. As shown in Table 1, Capsule Network presents the lowest accuracy and ISTVT performs better than STIL. Our approach performs much better than Capsule Network, and the in-dataset accuracy is a little bit higher than that of STIL and ISTVT.

Table 1. Deepfake detection accuracy. The best in-datasets accuracy results are bolded. Except for Capsule Network, which is reproduced, the other methods' results are from [10,28].

Methods	FF++				DFDC
	DF	F2F	FS	NT	
Capsule Net	98.5%	96.7%	94.5%	91.8%	-
STIL	99.6%	99.3%	97.1%	95.4%	89.8%
ISTVT	99.6%	99.6%	100%	96.8%	92.1%
Our approach	99.9%	99.6%	100%	98.8%	93.7%

DF: Deepfakes; F2F: Face2Face; NT: Neural Textures; FS: FaceShifter.

4.4. Ablation Test Results

To test the significance of the consistency branch of our model, we also conducted an ablation experiment (see Table 2) on Swin Transformer deepfake image classification with our approach (with consistency learning supervised). We found that consistency learning can effectively increase the accuracy, which is because our approach uses consistency loss to shorten the distances (increases the similarity) between different frames of the video so that if one frame is distinguished by the classification branch as fake with a really high confidence score, there is a high probability that other hard sample frames will be regarded as fake as well.

Table 2. Ablation experiment accuracy comparison. We utilized Swin Transformer to conduct a deepfake image binary test and compare its accuracy with that of our approach.

Methods	FF++(Binary Test)	DFDC
Swin Transformer	89.4%	91.4%
Our approach	94.3%	93.7%

In addition, the balance weight α is another variable that decides the final total loss; thus, we conducted an ablation experiment to determine how much weight is required for the consistency loss to show the best performance in this discriminative model by setting this hyper-parameter to 0.4, 0.5, 0.6, 0.7 and 0.8. In this ablation test, we still utilized FaceForensics++ as the dataset to explore what balance weight could reach the highest predicted accuracy. In addition, we combined all the fake categories into one class to test binary performance instead of testing multiple classifications. Because we did not set this weight balance as a learnable parameter in our model, we chose to traverse the weight balance by adding 0.1 per experiment. Once we obtained the highest accuracy, we regarded this model as the state-of-the-art model, and we utilized this model's trained parameters to conduct cross-dataset experiments. As shown in Table 3, we find that the weight balance setting of 0.7 shows the best performance, and was trained on the FaceForensics++ binary test.

Table 3. Ablation experiment accuracy on different weight balances. This is to test which weight balance of our loss function will achieve the best result.

α	0.4	0.5	0.6	0.7	0.8
Accuracy	92.6%	93.9%	94.1%	94.3%	93.7%

4.5. Cross-Dataset Study

Due to the multiple forgery methods, it is important to evaluate the model's cross-dataset performance. To test the generalization ability of our model, we trained our model on FaceForensics++ and tested it on Celeb-DF, FaceShifter and DFDC. Considering that most deepfake datasets' sample sizes for the real class and fake class are unbalanced, we

also chose to utilize the Area Under the Curve (AUC) to evaluate the robustness of our model, as shown in Table 4.

Table 4. The AUC of the cross-dataset experiment. We trained our model on FF++ and tested the unseen data. The best cross-dataset AUC results are bolded. The other methods' results are from [10,28–31].

Models	Celeb-DF	FaceShifter	DFDC	Average
Capsule Net	34.5%	42.4%	-	38.5%
STIL	75.6%	-	-	75.6%
ISTVT	84.1%	99.3%	74.2%	85.9%
Face X-ray	79.5%	92.8%	65.6%	79.3%
FTCN	86.9%	98.8%	74.0%	86.6%
LipForensics	82.4%	97.1%	73.5%	84.3%
Our approach	88.8%	98.3%	71.8%	86.3%

We found that our approach could only have the best results on Celeb-DF. This is because Celeb-DF contains advanced post-processing work such as color blending and it is really hard to find out when boundary mismatch happens in generated media. To prove this point of view, we chose to create heatmaps of our model and other CNN-based detection models to explain the model's visualization of focus regions of deepfake media. By observing the heatmap of our model, we found that our model focuses more on generated human facial contents such as lips and eyes instead of boundary texture information (see Figure 6). By checking some example videos of FaceShifter, we found that forgery clues always happen in the facial boundary area, which is the facial region that our model pays less attention to, but Celeb-DF's forgery clues occur on the facial features, such as unnaturally manipulated human eyes and lips. In addition, we compared the data in Celeb-DF with the data of DFDC, and we found that DFDC contains more data with different shooting scenarios, such as dark indoor backgrounds and faces with extreme poses. This makes it really hard for our model to achieve consistency and obtain useful information to distinguish real or fake media because we utilized the model trained by FaceForensics++, and it included more human faces with normal poses and bright shooting scenarios, which is similar to the Celeb-DF data distribution.

In addition, we also compared the Transformer-based method ISTVT as a benchmark model, and it achieved slightly better AUC results on FaceShifter and DFDC (see Figure 7). This is because the authors who used ISTVT applied a self-subtraction mechanism and temporal self-attention in their work, which is what our approach lacks. The self-subtraction mechanism can focus more on inter-frame distortion, which is another significant technique for deepfake detection. It requires the application of an input token before the projection to query and key vectors. It generates feature residuals to replace the original features. However, our model still achieves competitive results compared with other previous state-of-the-art models such as STIL and Face X-ray.



Figure 6. Comparison heatmap between other deepfake detector methods and our approach. From this figure, it is obvious that most deepfake detectors focus on boundary information of generated fake media.

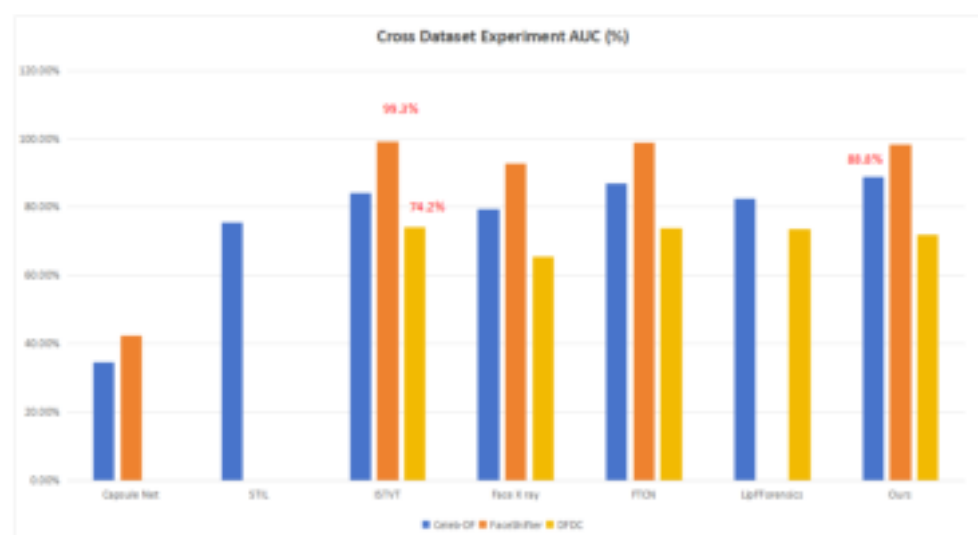


Figure 7. Cross-dataset AUC results based on Capsule Net, STIL, ISTVT, Face X-ray, FTCN, LipForensics and our approach. The best performance on each dataset is labeled in the figure.

5. Conclusions and Future Work

In this paper, we propose Swin-Fake, which, for the first time, combines Swin Transformer with consistency loss, as a consistency learning method for deepfake detection. We employ Swin Transformer as the feature extractor and utilize the average Cosine distance as a measure of consistency loss, which improved generalization ability in some deepfake databases. We proved that a Transformer-based classifier could reach a higher level in the deepfake detection field compared with other previous CNN-based methods. During the training and validation phases, we concluded that Swin Transformer had better deepfake distinguishing ability, while large-scale databases such as FF++ and DFDC do not exhibit promising performance on some small datasets such as Celeb DF. In addition, calculating the consistency between the frames' features could significantly decrease the loss and enhance the evaluation metrics; thus, we believe that applying the consistency learning method with Transformer for deepfake detection could be a new trend in fake media detection.

Deepfake video detection cannot be completed without temporal information because frame jittering always happens in the adjacent frames. In recent research, temporal attention is always combined with spatial attention for video detection. As another technique in the video detection field, temporal attention can be seen as a dynamic temporal selection mechanism that determines when to pay attention to a certain feature of a video, and is often applied in video processing. It is also an effective method to increase the evaluation

metrics in this field if we can combine temporal consistency information together with extracted spatial information, and the most common method is to design a specific modules to explore temporal consistency or inconsistency by applying horizontal and vertical convolutional layers. Thus, in future work, we will further investigate temporal self-attention of the Swin Transformer-based backbone. Even though our current work is focused on video-based detectors, we still focus on spatial feature extraction, and feature consistency is applied for intra-class comparison. Combining temporal consistency with spatial consistency would probably involve more information, which would benefit the classification of videos. As for the weight balance α , we added 0.1 in each experiment to test the efficiency of the consistency block and obtain the value with the highest accuracy. To obtain a more accurate value of this hyper-parameter, we will set the weight balance as a learnable parameter in future investigations. Additionally, we will further investigate other visualization methods for Swin Transformer-based deepfake detectors to interpret what spatial features are captured and contribute to the deepfake classifier.

Author Contributions: Conceptualization, L.Y.G. and X.J.L.; methodology, L.Y.G. and X.J.L.; software, L.Y.G.; validation, L.Y.G.; formal analysis, L.Y.G.; investigation, L.Y.G.; resources, L.Y.G. and X.J.L.; data curation, L.Y.G. and X.J.L.; writing—original draft preparation, L.Y.G., X.J.L. and P.H.J.C.; writing—review and editing, L.Y.G., X.J.L. and P.H.J.C.; visualization, L.Y.G. and X.J.L.; supervision, X.J.L. and P.H.J.C.; project administration, X.J.L. and P.H.J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The relative datasets are downloaded from [24], DFDC reproduced codes are from [13].

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Wared-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Network. *Commun. Acn* **2018**, *63*, 139–144. [CrossRef]
- Karras, T.; Laine, S.; Aila, T. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
- Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. In Proceedings of the 34th International Conference on Neural Information Processing System, Red Hook, NY, USA, 6–12 December 2020; pp. 6840–6851.
- Nguyen, H.H.; Yamagishi, J.; Echizen, I. Use of a Capsule Network to Detect Fake Images and Videos. In Proceedings of the Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019.
- Das, S.; Seferbekov, S.; Datta, A.; Islam, M.S.; Amin, M.R. Towards solving the deepfake problem: An analysis on improving deepfake detection using dynamic face augmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021.
- Zhao, T.; Xu, X.; Xu, M.; Ding, H.; Xiong, Y.; Xia, W. Learning Self-Consistency for Deepfake Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, Virtual, 26 April–1 May 2020.
- Khan, S.A.; Dai, H. Video Transformer for Deepfake Detection with Incremental Learning. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual Event, 20–24 October 2021; pp. 1821–1828.
- Khormali, A.; Yuan, J. DFDT: An End-to-End DeepFake Detection Framework Using Vision Transformer. *Appl. Sci.* **2022**, *12*, 2953. [CrossRef]
- Zhao, C.; Wang, C.; Hu, G.; Chen, H.; Liu, C.; Tang, J. ISTVT: Interpretable Spatial-Temporal Video Transformer for Deepfake Detection. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 1335–1348. [CrossRef]
- Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021.
- Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; Yang, Y. Random Erasing Data Augmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
- DFDC Selim. Available online: https://github.com/selimsef/dfdc_deepfake_challenge (accessed on 15 May 2024).

14. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
15. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
16. Ni, Y.; Meng, D.; Yu, C.; Quan, C.B.; Ren, D.; Zhao, Y. CORE: Consistent Representation Learning for Face Forgery Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
17. Sun, K.; Yao, T.; Chen, S.; Ding, S.; Li, J.; Ji, R. Dual Contrastive Learning for General Face Forgery Detection. *AAAI Conf. Artif. Intell.* **2022**, *36*, 2316–2324. [[CrossRef](#)]
18. Zhang, J.; Cheng, K.; Sovernigo, G.; Lin, X. A Heterogeneous Feature Ensemble Learning based Deepfake Detection Method. In Proceedings of the ICC 2022—IEEE International Conference on Communications, Seoul, Republic of Korea, 16–20 May 2022; pp. 2084–2089.
19. Rana, M.S.; Sung, A.H. DeepfakeStack: A Deep Ensemble-based Learning Technique for Deepfake Detection. In Proceedings of the 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), New York, NY, USA, 1–3 August 2020; pp. 70–75.
20. Ilyas, H.; Javed, A.; Malik, K.M. AVFakeNet: A unified end-to-end Dense Swin Transformer deep learning model for audio–visual deepfakes detection. *Appl. Soft Comput.* **2023**, *136*, 110124. [[CrossRef](#)]
21. Khalid, F.; Akbar, M.H.; Gul, S. SWYNT: Swin Y-Net Transformers for Deepfake Detection. In Proceedings of the 2023 International Conference on Robotics and Automation in Industry (ICRAI), Peshawar, Pakistan, 3–5 March 2023; pp. 1–6.
22. Guo, J.; Deng, J.; Lattas, A.; Zafeiriou, S. Sample and Computation Redistribution for Efficient Face Detection. In Proceedings of the International Conference on Learning Representation, Virtual Event, 3–7 May 2021.
23. Zhou, T.F.; Wang, W.G.; Liang, Z.Y.; Shen, J.B. Face Forensics in the Wild. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021.
24. Kaggle. Available online: <https://www.kaggle.com/c/deepfake-detection-challenge/overview> (accessed on 12 December 2023).
25. Li, Y.Z.; Yang, X.; Sun, P.; Qi, H.G.; Lyu, S. Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics. In Proceedings of the Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020.
26. Li, L.; Bao, J.; Yang, H.; Chen, D.; Wen, F. FaceShifter: Towards High Fidelity And Occlusion Aware Face Swapping. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
27. Loshchilov, I.; Hutter, F. Decoupled weight decay Regularization. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
28. Gu, Z.; Chen, Y.; Yao, T.; Ding, S.; Li, J.; Huang, F.; Ma, L. Spatiotemporal Inconsistency Learning for Deepfake Video Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021.
29. Li, L.; Bao, J.; Zhang, T.; Yang, H.; Chen, D.; Wen, F.; Guo, B. Face X-ray for More General Face Forgery Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
30. Zheng, Y.; Bao, J.; Chen, D.; Zeng, M.; Wen, F. Exploring temporal coherence for more general video face forgery detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 15044–15054.
31. Haliassos, A.; Vougioukas, K.; Petridis, S.; Pantic, M. Lips don't lie: A generalisable and robust approach to face forgery detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 5039–5049.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

A Prosumer Hydro Plant Network as a Sustainable Distributed Energy Depot

Michał Morawski  and Przemysław Ignaciuk * 

Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology, 93-005 Lodz, Poland; michal.morawski@p.lodz.pl

* Correspondence: przemyslaw.ignaciuk@p.lodz.pl

Abstract: The shortage of efficient, low-cost storage depots inhibits the large-scale adoption of volatile-by-nature, renewable sources of energy (RSEs). In this paper, we outline how to utilize prosumer-owned hydro plants of a few to several kW as a distributed, short-term energy storage solution that is deployable with little investment and a low operational expenditure. The proposed solution is a system of interconnected hydro depots with an active water-flow control algorithm that reduces the grid's load variability and benefits prosumers. According to the tests conducted, prosumer revenue grows from several percent to over 30 percent, depending on weather conditions, in comparison to the free-flow case. In turn, the cushioning effect of the distributed energy buffer balances the fluctuations introduced by other RSEs, e.g., photovoltaic- or wind-based ones. Hence, while benefitting the involved parties, it also facilitates the inclusion of RSEs within the power distribution system.

Keywords: energy depot; hydro plants; distributed systems



Citation: Morawski, M.; Ignaciuk, P. A Prosumer Hydro Plant Network as a Sustainable Distributed Energy Depot. *Electronics* **2024**, *13*, 3043. <https://doi.org/10.3390/electronics13153043>

Academic Editors: Temitayo Olowu and Raghav Khanna

Received: 24 June 2024

Revised: 22 July 2024

Accepted: 30 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Electrical energy is considered the bloodstream of today's society. Electricity use in homes, industry, transportation, and building cooling and heating results in increased demand on a daily basis. Meanwhile, energy generation continues to be expensive, both from an environmental and an economic point of view. Global pollution, the growing cost of fossil fuels, and political pressure are just a few factors that encourage energy production from renewable sources (RSEs), e.g., the sun, wind, waves, geothermal energy, and water movement [1]. Unfortunately, the amount of energy retrieved from RSEs depends heavily on temporal weather conditions. RSEs provide inexpensive energy either at random hours (wind) or around noon (solar, mostly photovoltaic (PV)). In contrast, the demand remains at a similar level until the evening, as illustrated in Figure 1. Hence, costly, fossil-fuel-sourced plants need to be engaged in the evenings, but their activity should be constrained in favor of RSEs if the sun or wind is available. Moreover, energy generation from traditional thermal plants cannot be constrained too much due to technological factors. As a result of market dynamics, energy prices are set low (or even negative) at certain times to then reach a higher order of magnitude a few hours later, resembling the 'duck curve' depicted in Figure 2, which shows the initial days of July 2023 in the Polish energy market. In the interest of all stakeholders, this disparity should be reduced.

A self-imposed solution to the producer-consumer disparity is to store surplus energy when it is cheap for a later time when costly sources are incorporated. Unfortunately, today's technology does not permit executing this on a larger scale [2]. For instance, pumped hydro plants in Poland can deliver, at most, 5% of the overall power demand, and the battery depots only deliver 1% at most. Both of these traditional methods, as well as prospective hydrogen-based systems, are expensive to deploy and maintain. The low durability of electromechanical batteries further restricts a good return on one's investment. Hence, seeking a new, low-CAPEX (capital expenditures) and low-OPEX (operational expenditures) option for energy storage is desirable.

Recently, a substantial change in the electricity market has been observed. Traditional, large power plants are being supplemented by small-scale facilities installed by citizens on their own premises. These citizens are electricity consumers during certain hours of the day and producers at other times and are thus called prosumers. On the one hand, the popularity of prosumer installations advances the adoption of RSEs; on the other hand, it threatens the stability of the energy supply system by forcing extreme load changes on traditional plants.

This study contributes to the aforementioned challenges in the following ways:

- We propose engaging prosumer hydro plants with a peak power in the range of a few kW in order to form a distributed energy depot;
- We design an active control scheme, which allows for both increasing the revenue of the plant owners and reducing the load variability imposed on legacy thermal plants (“beheading the duck” [3]), thus achieving a win–win solution;
- Contrary to the typical approach, which focuses on a single hybrid generator, e.g., [4], we show how to co-ordinate a network of plants for a safe and efficient distributed system operation.

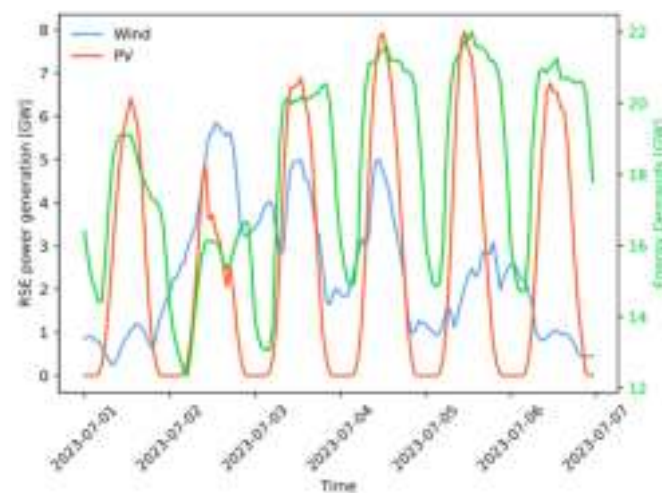


Figure 1. Energy generation from principal RSEs and predicted energy demands in Poland for 1–6 July 2023 (1 July was a Saturday [5]).

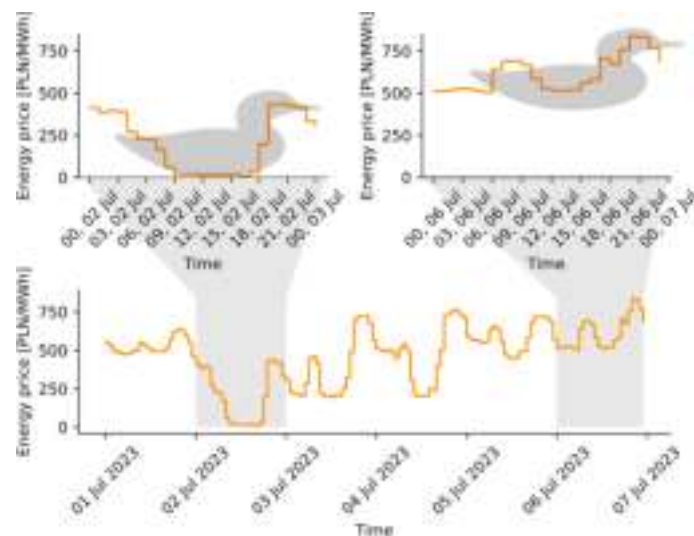


Figure 2. Fluctuation in next-day market energy prices [PLN/MWh] in the Polish market over 1–6 July 2023, following the ‘duck curve’ [5].

The remainder of the paper is organized as follows. In Section 2, the traditional methods of hydro plant control are discussed. In Section 3, we elaborate on the differences between conventional and prosumer plants. In Section 4, a discrete-time dynamic model of water flow in a multi-hydro plant system is constructed, and a procedure to optimize its operation is described. In Section 5, we evaluate the algorithm's performance for a broad spectrum of test data. Then, we discuss the implementation considerations in Section 6. Finally, conclusions are drawn in Section 7.

2. Control of Hydro Plants

In the literature, hydro plants (HPs) are classified as normal (>100 MW), medium (<100 MW), small (<15 MW), mini (<1 MW), micro (<100 kW), or pico (<5 kW) HPs, although the actual values vary by region [6,7]. Most of the research concentrates on small, medium, and normal HPs, which play a pronounced role in electricity supply. In this paper, owing to their significance for prosumers, micro- and pico-HPs are investigated. They encompass numerous formerly abandoned mills with small artificial ponds, swamps, and other natural formations, commonly known as run-of-river HPs [8,9], thus requiring insignificant capital expenditures for deployment. For example, currently, in Poland, less than 5% of possible installations are used for energy production [10]. Moreover, numerous online service publications show that pico-HPs can be easily deployed without specialized equipment [11]. Thus, the target area of the study conducted in this work is important. Of course, a single prosumer depot is not a game-changer. However, a revenue-oriented, optimally scheduled set of prosumer HPs may constitute an efficient energy storage system resistant to perturbances with low operational costs. In fact, similar circumstances led to the popularity of PV installations, which have become a common element of the present landscape.

However, there is a fundamental difference between PV and HP installations. The PV ones are mutually independent, whereas the proper operation of an HP depends on all the stations upstream being linked in a cascade. In the past, the scheduling of the operation of a cascade of HPs has been investigated as a case study [12] rather than as a networked system subjected to a control action, i.e., each plant was adjusted manually without enforcing a common framework. While a suitable scheduling plan can be defined in this way, it may not account for the optimal operating conditions or pose a risk in terms of drought/flood protection. Other approaches consider simplified models, reducing the cascade to a single-plant case [4], where the dams are at a close distance [8,13] or located on one river but not on its tributaries [9,14]. A solution conceptually closest to the one discussed here has been proposed in [13], albeit it targets large, concentrated installations and not the distributed HPs under prosumer management.

Technically, the steering of energy production at a hydro plant is not an intricate process. By raising the weirs, one constrains the water flow and, by lowering them, one increases energy generation. However, establishing the correct sequence of raising and lowering the weirs is far from obvious. Usually, the scheduling strategy is categorized as long-term (1–5 years), mid-term (3–18 months), and short-term (2–14 days) [15]. Nevertheless, in the case of prosumer HPs, such strategies are of limited usefulness due to the small volume of the ponds. In the considered case, a meaningful scheduling strategy should span the range of a few hours or, at most, one day. Moreover, it should focus on decreasing the generation–consumption imbalance in the evenings and mornings (Figure 2) or, equivalently, take advantage of the fluctuating market prices resulting from the disparity between forecasted and actual demands [16] (Figure 1).

The optimal scheduling of the HP water release problem has been investigated for a long time ([15,17] and the references therein). However, in those studies, HPs were examined only in the context of either flood/drought protection or power generation. Here, a third role of HPs is explored, i.e., the potential to provide an energy depot [2]. Instead of focusing just on the amount of generated electricity, HPs may constrain the river flow when the energy is cheap and increase it when the price grows, thus creating an efficient

energy buffer. The overall amount of the generated energy will remain the same, but the HP owner revenue will grow by embracing the short-time changes in the price profile. Simultaneously, the fluctuation of energy generation from the traditional thermal plants decreases. A similar idea has been explored in [18], but it is destined for large installations, not the prosumer ones considered in this work.

3. Prosumer Hydro Plants

Not all RSEs are sensitive to the influence of weather fluctuations, e.g., geothermal, tide-based, or large, costly hydro HPs [19]. Here, the focus is on small, flexible installations, i.e., micro- [6] and pico-HPs [20]. They are typically deployed as run-on-river installations [8,21], i.e., without large dams and reservoirs, and thus they do not have significant expenditures or intrusion into the environment [21]. As opposed to larger HPs, which are not affected by local atmospheric conditions since the amount of water over the predictions is negligible compared to the reservoir volume, the limited storage space of micro- and pico-reservoirs does make them susceptible to weather variability. For instance, a local storm can promptly fill up the pond near one micro-HP while a nearby one simultaneously has sunny and dry conditions. Today, precipitation forecasting is not precise enough to deal with such situations. The water runoff models are not accurate either [22]. Therefore, when the weather is unstable, the plant schedule has to be recomputed frequently to adjust the system to the most recent circumstances or the price variability in a real-time, balancing market [16]. Contrary to the commonly used numerical optimization approaches, the algorithm presented in this manuscript takes a fraction of a second to execute, even on a low-end device.

A set of HPs uses the water from upstream plants and tributaries. Hence, improper control may breed a flood. Therefore, it is mandatory to synchronize the HP operation, explicitly taking into account diverse factors:

- (1) Geographical factor—the distance between the plants and the ensuing delay of water reflow;
- (2) Weather-related factor—the predicted precipitation level and uncertainty level;
- (3) Application-oriented factor—the use of the HP cascade as a distributed energy depot combined with the common function of power generation.

Usually, the control schedule is established using numerical optimization procedures, incorporating artificial intelligence techniques to tackle the computational burden [15,17,21–29]. However, these procedures exhibit robustness issues in the considered application area, even in relatively simple cases [29]. Moreover, the large amount of time that one needs to arrive at a solution (using reasonable computing resources) contradicts the need for plan adaptability, i.e., introducing changes to the control schedule when the weather is unstable or a malfunction occurs.

To capture factors (1)–(3), information about different delays in the water flows on the links connecting the reservoirs is explicitly incorporated into the dynamical model constructed in this work. Using the system's dynamical representation, an optimization problem is stated. It may be solved analytically if saturation is absent [30]. In contrast, here, the natural constraints are embedded in the algorithmic solution. This proposal brings benefits not only to prosumers by increasing their economic gain but also to the power grid operators by reducing the load variation for the standard power plants.

4. System Model

Usually, HP modeling concentrates on the optimization of the work of generators, ignoring the supply system [23]. Here, the generator is considered a black box, and the focus is placed on the connected HPs as a water supply system. A key point to consider in a water reflow system is the non-negligible time between issuing the control action at one reservoir before it influences the water level at a downstream one. Therefore, as opposed to the earlier models of storage networks, e.g., [31], in the approach advocated here, the control principles from time-delay storage systems [32,33] will be applied. However, the models

proposed in [32,33] assume continuous-time control adjustment, which is challenging to realize in a water control system owing to the specifics of the mechanical components steering the dam weirs. The model in this work explicitly covers the effects of a finite sample time and will be constructed directly in the discrete-time domain.

4.1. Single-Plant Depot

Let us consider the model of a single HP as a constituent element of the system illustrated in Figure 3. The water budget dynamics at the plant may be described via the recursive relation

$$s_j(k+1) = s_j(k) - f_j(k) + \sum_{\substack{i \in \text{plants} \\ \text{upstream}}} f_i(k - T_{ij}) + r_j(k), \quad (1)$$

where

- $k \in [1, m]$ is a time instant and m is a scheduling horizon;
- $s_j(k)$ is the amount of water stored in the reservoir near plant j (in Figure 3, plant $j = 3$ is shown in an enlarged view); $s_j(k)$ encompasses hydroengineering construction as well as nearby swamps and similar areas;
- $f_j(k)$ is the amount of water used to drive the power generators installed at dam j in the time between instants k and $k + 1$;
- $r_j(k)$ is the supply from external hydrological sources like rain (and its runoff), melting snow, uncontrolled tributaries, and vaporization. The $r_j(k)$ values can be obtained from the weather forecast and hydrological models within the planning horizon of m periods. $r_j(k)$ may be positive (prevalence of precipitation) or negative (prevalence of vaporization). $r_j(k)$ is assumed to be known in the model. When the true value of $r_j(k)$ differs from the forecast, the algorithm is run again, which does not involve complex computations, as discussed in Section 4.3.

The tributaries supply the reservoir with the water previously used by the plants upstream. The water from upstream plant i arrives at plant j with $T_{ij} > 0$ delay. The period length Δk , i.e., the time between instants k and $k + 1$, can be selected arbitrarily, but according to the pace of price changes, it is reasonable to choose 1 h down to 15 min, depending on local regulations. Similarly, the planning horizon m may cover a 24 h window of known energy prices (the next-day market). The initial flow $f_j(k \leq 0)$ and the initial water level $s_j(0)$ are assumed to be known. The terminal condition $s_j(m)$ can be selected arbitrarily.

The income from the plant obtained in one period may be calculated as

$$J_j(k) = \eta_j p_j(k) f_j(k) \Delta k, \quad (2)$$

where

- $p_j(k)$ is the energy price at instant k . The fact that all the HPs are connected to the same system does not induce the same price profile for each plant j . The owners may have a contract with different companies, use different tariffs, or have different consumption patterns themselves;
- η_j is the efficiency of power generators, incorporating the impact of the dam height. For the prosumer generators in the lowlands, the flow of $1 \text{ m}^3/\text{s}$ corresponds to the power generation of 5–7 kW.

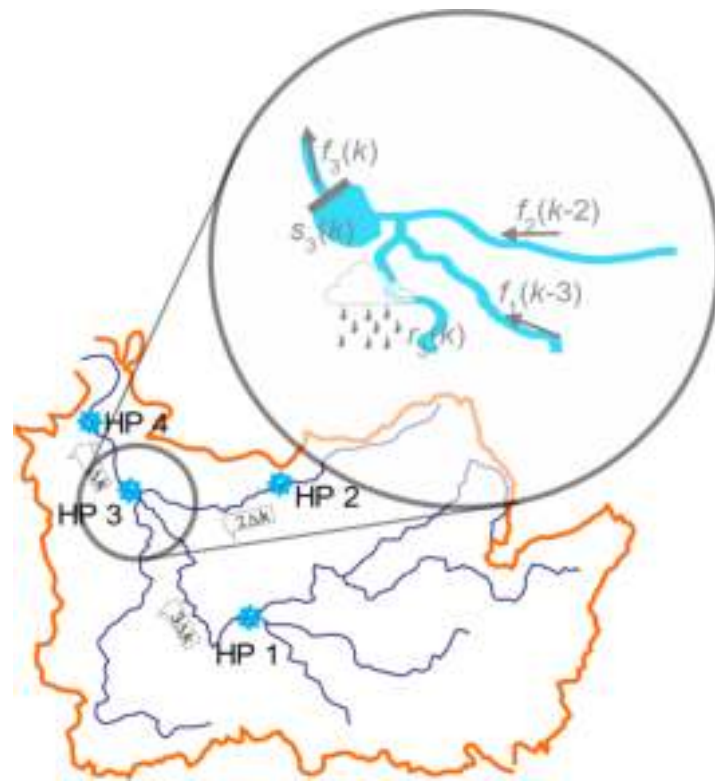


Figure 3. Model of connected hydro plants in a river basin. Plant 3 (magnified) is supplied by the inflows from reservoirs 1 and 2 with delays indicated by thick arrows (Δk —sampling period) and exogenous sources, e.g., rain or tributary in the south. The amount of water delivered to the plant from these sources within Δk at instant k is $f_1(k-3)$, $f_2(k-2)$, and $r_3(k)$, respectively. The outflow from the dam generators at plant 3 is $f_3(k)$. The input and output flow difference is buffered in the pond, whose current water volume is denoted by $s_3(k)$.

4.2. Multi-Plant Depot

Similar to PV, as the number of HPs grow, their impact on the grid increases. A collection of pico- and micro-HPs may constitute a large, distributed energy depot.

In the case of n power plants in the same river basin, the model variables are grouped into vectors:

$$\mathbf{s}(k) = \begin{bmatrix} s_1(k) \\ s_2(k) \\ \vdots \\ s_n(k) \end{bmatrix}, \quad \mathbf{f}(k) = \begin{bmatrix} f_1(k) \\ f_2(k) \\ \vdots \\ f_n(k) \end{bmatrix}, \quad \mathbf{r}(k) = \begin{bmatrix} r_1(k) \\ r_2(k) \\ \vdots \\ r_n(k) \end{bmatrix}, \quad (3)$$

where $\mathbf{s}(k)$ is the vector of reservoir water level, $\mathbf{f}(k)$ is the vector of water volumes in inter-reservoir flows, and $\mathbf{r}(k)$ is the vector of water volumes from exogenous sources, respectively.

Derived from (1), the networked system dynamics are described by

$$\mathbf{s}(k+1) = \mathbf{s}(k) + \sum_{t=0}^T \Theta_t \mathbf{f}(k-t) + \mathbf{r}(k), \quad (4)$$

where T is the maximum delay, and matrices Θ_t group the information about the geographical topology and flow delays. $\Theta_t = [\theta_{ij}]_{n \times n}$, with $\theta_{ij} = 1$, if the flow from reservoir j reaches reservoir i with delay t , and otherwise, it is 0. Θ_0 describes the outflow from the system. Contrary to [13], here, the distance between plants is non-negligible. For the example from Figure 3, the longest delay $T = 3$ (the flow between reservoirs 1 and 3) and

$$\Theta_0 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \Theta_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \Theta_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \Theta_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (5)$$

For the multi-plant system, the quality index (2) becomes

$$J = \sum_{k=1}^n J_j = \sum_{k=1}^n \sum_{j=1}^m J_j(k) = \sum_{k=1}^n \sum_{j=1}^m \eta_j p_j(k) f_j(k) \Delta k. \quad (6)$$

The stakeholders tend to have contrasting objectives. Prosumers try to maximize J_j , whereas the grid operators want to maximize J . A convenient way to reconcile their interests is a price system. The algorithm presented in the latter part of this paper satisfies the objectives of both plant and grid owners.

4.3. Control Algorithm

The main challenge to solving problem (6) relates to the complexity of system dynamics originating from the presence of non-negligible delay in (4). An additional difficulty stems from the physical restrictions imposed on the variables $\mathbf{f}(k)$ and $\mathbf{s}(k)$

$$\begin{aligned} \forall_{j,k} f_j^{\min} &\leq f_j(k) \leq f_j^{\max}, \\ \forall_{j,k} s_j^{\min} &\leq s_j(k) \leq s_j^{\max}, \\ \forall_j s_j(m) &= S_j, \end{aligned} \quad (7)$$

where S_j is an arbitrarily chosen terminal value.

As stated above, the standard solvers are not robust in the class of problems considered here [15,17,21–29]. Therefore, to address the computational aspects, a three-step min–max optimization procedure sketched in Figure 4 is applied. Each step of the algorithm is explained below. The communication among HPs conforms to the well-known industry standard Message Queuing Telemetry Transport (MQTT) [34], which provides both security and synchronization among the plant controllers.

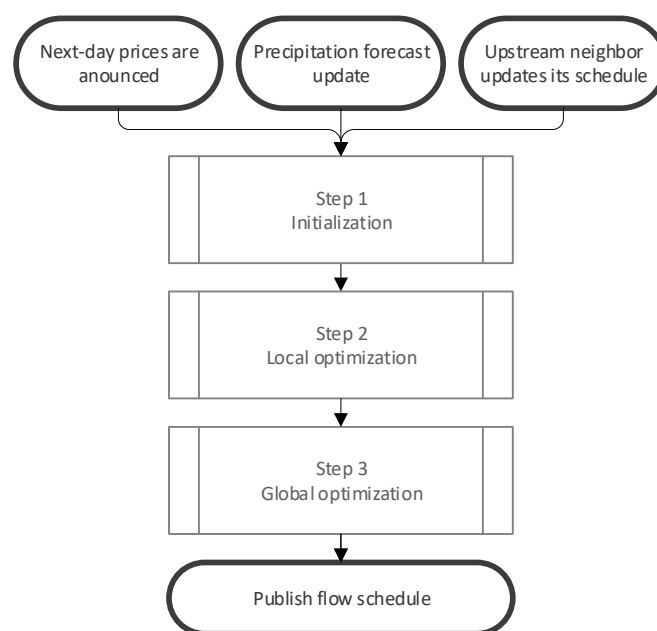


Figure 4. Algorithm flow chart.

Step 1, initialization:

Input: price profile.

Output: preliminary water flow schedule.

Objective: Determine the intervals of accumulation and earning within the considered time frame, as shown in Figure 5. When the energy prices are above the average, then it is time to earn money (earning interval). Otherwise, one should store the water (accumulation interval). In the first case, $f(k) = f^{\max}$, and in the latter case, $f(k) = f^{\min}$ for each k in the respective interval. As a result, for plant j , h_j periods with length m_{jp} are obtained. Since the HPs are managed independently owing to different tariffs or demands, the segmentation pattern may differ for each HP. Either way, this step does not involve complex computations. The approximation obtained in Step 1 may result in crossing the bounds of $s(k)$ defined in (7).

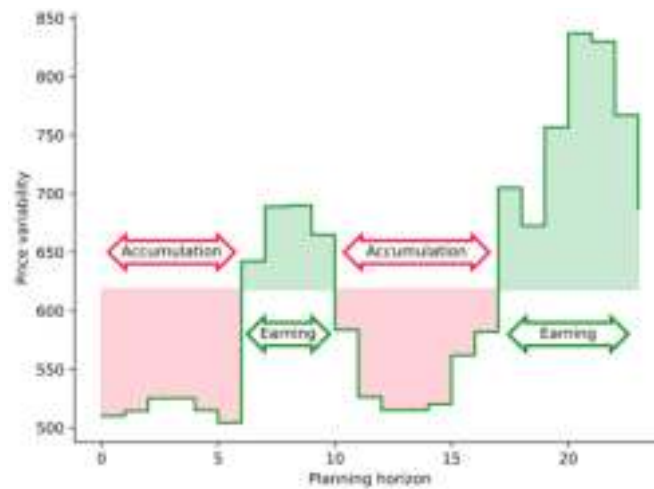


Figure 5. Illustration of Step 1: identifying accumulation and earning intervals for the price profile from 6 July. Marked periods impact the version of Step 2 of the presented algorithm.

Step 2, local optimization:

Input: price profile and preliminary water flow schedule computed at Step 1, as well as the inflow (i.e., predicted rain, tributaries inflow, the schedule of upstream plants) to the system.

Output: locally optimized flow schedule. The result does not cross the limits defined in (7).

Objective: Looking for the optimal flow within each period found by execution of Step 1.

Since two kinds of periods exist, it is reasonable to split Step 2, accordingly.

Step 2a (accumulation interval): If, for any k , $s_j(k) > s_j^{\max}$, it means that the water flow is constrained too much. Thus, the inflow of water floods the pond. Therefore, it is desirable to find instant k^* with the highest price within interval n_j and increase the flow by $s_j(k^*) - s_j^{\max}$, but not above $f_j^{\max}$. This step is repeated until the condition $s_j(k) \leq s_j^{\max}$ is satisfied for all k in the investigated interval.

Step 2b (earning interval): If, for any k , $s_j(k) < s_j^{\min}$, it means the flow is excessive. Thus, one should find an instant k^* with the lowest price in this interval and constrain the flow by $s_j^{\min} - s_j(k^*)$ but not below $f_j^{\min}$. This step is repeated until the condition $s_j(k) \geq s_j^{\min}$ is satisfied for all k in the investigated interval.

Step 2 is executed for each HP, starting from the upstream ones. The maximum number of iterations is $\sum_p m_{jp}^2$ for each HP j , which is low enough to perform in real-time, even for a large-scale system. Step 2 is illustrated in Figure 6. After computations, the maximum possible gain is obtained within each interval.

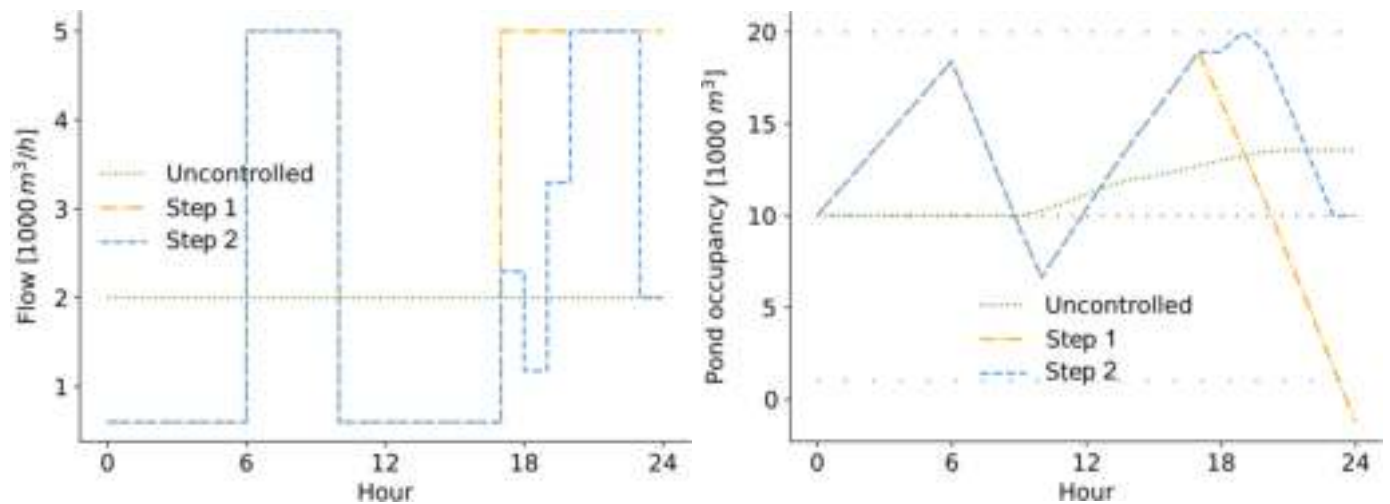


Figure 6. Flow intensity obtained in Step 2 for the pattern from Figure 5. No action is required for the first 3 periods. In the last period, terminal condition $s(m)$ is violated, so the flow needs to be constrained.

Step 3, global optimization:

Input: price profile and locally optimized water flow schedule computed at Step 2.

Output: globally optimized flow schedule.

Objective: Rescheduling water flow within the whole optimization horizon.

Step 2 results in the optimal flow within each earning/accumulation interval and guarantees that condition (7) is satisfied for each interval n_j . Possibly, there is room for globally boosting the system's performance. By observing the events illustrated in Figure 6, one may note that in the hours 0–19, the pond occupancy grows nonmonotonically until the reservoir fills up. The filling and draining intervals are delineated by the instants when the pond saturates. The rescheduling inside these intervals does not influence the final volume of reservoirs, so it is possible to increase the flow when the energy is expensive and decrease it by the same value when the energy price is lower.

The maximum number of iterations in Step 3 is similar to those in Step 2. The result of an example Step 3 is shown in Figure 7.

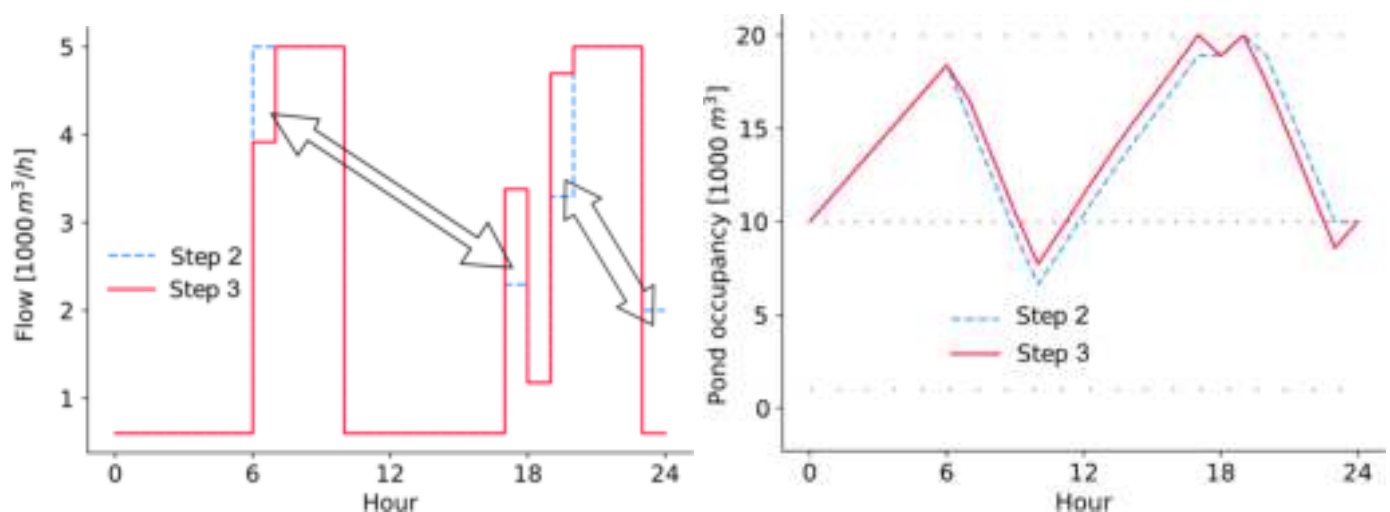


Figure 7. Outcome of Step 3 juxtaposed with the outcome of Step 2 (Figure 6). Arrows mark the rescheduled water volume.

Final considerations

The algorithm is executed whenever the input values change, e.g., when new energy prices are announced or a weather forecast update $r_j(k)$ is obtained. The results must be broadcast to all the concerned owners in the HP's network because Step 2 of the algorithm needs the output from upstream plants. The algorithm can be run locally using inexpensive hardware, e.g., a microcontroller.

5. Evaluation

The four-node topology in Figure 3 was used for the algorithm assessment. It is simple enough to illustrate the algorithm's properties while covering the major challenges of a prosumer HP network originating from arborescent topology and non-negligible delay among the plants. No common management is assumed—the prosumers make the control decisions at their plants independently from each other. Only broadcasting the control decisions is necessary. Formal regulations can easily enforce it.

The algorithm from Section 4 was tested using diverse price profiles taken from [5], including those illustrated in Figure 2, in the presence of a moving wave of intensive rain and the corresponding runoff depicted in Figure 8. The computations were executed for each profile and different volumes of ponds. In all the cases, an exogenous flow of constant intensity $f(k) = [2, 3, 5, 5] \times 10^3$ [m³/h] supplies the system. Other system parameters are taken as $f^{\max} = [5, 6, 10, 10] \times 10^3$ [m³/h], $f^{\min} = [0.5, 0.6, 1, 1] \times 10^3$ [m³/h], $s(0) = s(m)$, $s^{\min} = 0.1 \times s(0)$, $s^{\max} = 2 \times s(0)$, with the initial water level $s(0)$ set as follows:

Scenario 1 (small-size reservoirs): $s(0) = s(m) = [0.5, 0.75, 1.25, 1.25] \times 10^4$ [m³];

Scenario 2 (medium-size reservoirs): $s(0) = s(m) = [1, 1.5, 2.5, 2.5] \times 10^4$ [m³];

Scenario 3 (large reservoirs): HPs are connected to larger artificial lakes with a negligible probability of saturation, $s(0) = s(m) = [2, 3, 5, 5] \times 10^4$ [m³].

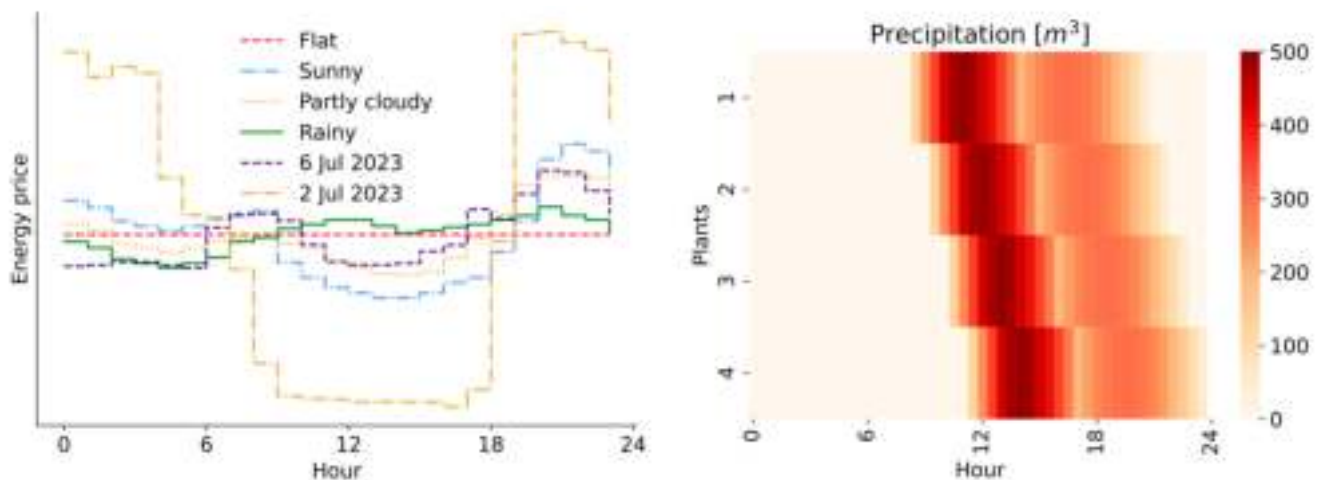


Figure 8. Example price profiles scaled to the same average and assumed precipitation pattern with runoff used to validate results.

While the algorithm does not expect any specific relation among the price profiles, for the sake of readability, all the HPs share the same data. For that reason, the turbine output is also taken as a linear function of the flow intensity $\eta_j(f_j(k)) = \text{const.}$

Table 1 summarizes the prosumer's gain and the grid operator's overall payoff. The gain is established with respect to the case when the HPs are not controlled. The discrepancy in the final state under free flow is refilled or released at the average price.

Table 1. Profit gain [%] with respect to free flow [m^3] for the price profiles from Figure 7 and different reservoir sizes.

Profile	Plant 1	Plant 2	Plant 3	Plant 4	All Plants	Remark
Small-volume ponds (Scenario 1)						
Sunny	10.85	11.80	10.98	6.74	9.71	
Partly cloudy	4.52	4.80	6.95	1.69	4.44	
Rainy	2.54	2.41	6.62	198.52	34.55	When Plant 4 is left uncontrolled, then the production drops below the minimum for 5 h.
6 July 2023	10.13	10.35	13.39	28.16	17.30	
2 July 2023	29.25	33.81	59.53	46.64	47.30	
Medium-volume ponds (Scenario 2)						
Sunny	20.13	19.38	15.91	11.44	15.65	
Partly cloudy	11.98	12.07	13.86	38.17	21.29	When Plant 4 is left uncontrolled, then the production drops below the minimum for 3 h.
Rainy	9.31	11.37	17.44	65.61	33.05	When Plants 3 and 4 are left uncontrolled, then the production drops below the minimum for 5 h (both), and Plant 3 experiences a flood during 2 h.
6 July 2023	14.47	13.77	14.25	49.92	25.15	When Plant 4 is left uncontrolled, then the production drops below the minimum for 4 h.
2 July 2023	68.50	68.34	69.79	82.83	74.03	When Plant 4 is left uncontrolled, then the production drops below the minimum for 2 h.
Large-volume ponds (Scenario 3)						
Sunny	21.04	20.82	19.52	16.15	18.86	
Partly cloudy	14.73	14.72	14.43	19.43	16.76	When Plant 4 is left uncontrolled, then the production drops below the minimum for 1 h.
Rainy	6.93	6.79	6.61	5.78	6.41	
6 July 2023	15.39	15.18	13.83	12.36	13.82	
2 July 2023	59.52	56.83	80.13	65.47	67.83	

The data shown in Table 1 reveal that the larger the pond volume, the more profit obtained. However, with the proposed algorithm applied, even for small reservoirs, the relative monetary benefit reaches several dozen and grows with price variability, albeit not linearly. The data also show that small changes in $r_j(k)$ or $p_j(k)$ may lead to a substantial discrepancy in the outcome. The gain grows when the distance between the HPs increases, which is attributed to a larger difference between $p_j(k)$ and $p_j(k - T_{ij})$. It is worth noting that the algorithm tends to fill up the ponds at low prices around midnight yet does not allow for floods. This behavior is advantageous as the ponds contribute to the soil moisture, thus alleviating the problem of persistent droughts in various regions.

The algorithm opts for the maximum or minimum flow intensity $f(k)$ rather than intermediate ones. This property facilitates turbine construction and maintenance by increasing efficiency in the nominal conditions.

The results obtained using the proposed algorithm, summarized in Table 1, are much better than those obtained by standard optimization algorithms [29]. Moreover, all the downstream HPs are protected from otherwise common floods and droughts (Scenario 2).

The balancing market may earn additional profit for both the prosumers and grid operators [16]. When the grid operator to which the prosumer network is connected detects a lack of energy, it may trigger an immediate water release or, conversely, constrain the flow in the case of surplus.

6. Implementation Aspects

Prior to deployment, a few more aspects should be considered:

1. Legal issues. System deployment requires consent from the regulator of water resources and corresponding agreements from the grid operators.
2. Information exchange. The designed algorithm requires a reliable communication platform among prosumer plants. We propose MQTT [35] for this purpose.

3. Weather forecast. Next-day forecasting is currently not accurate enough to predict local storms several hours ahead. However, radar-based imaging provides thorough precipitation estimates in a timeframe of a few hours.
4. Depots as a real-time system. Each HP's controller operates independently from others. Hence, the system is potentially prone to unfavorable phenomena from distributed systems like rushes or deadlocks. The MQTT protocol solves this issue without increasing computational load as the schedule calculation takes only a fraction of a second, and the forecasting updates are infrequent.
5. Depots as a control system. The system operates in an open loop. The updates to the schedule only impact the downstream HPs. The system is bounded-input bounded-output (BIBO) stable.
6. Security. The proposed communication platform conforms to high industry security standards [35].
7. Computational resources. The proposed algorithm does not require high-end computing resources to execute. It operates and communicates efficiently on low-end, inexpensive devices.

7. Conclusions

This paper presents an optimal strategy to control a system of connected HPs, forming a sustainable, distributed energy depot. Such a system benefits both the power grid operators by reducing fluctuations in the grid load and the plant owners by increasing their income. It can be deployed with low capital and operational expenditures. In addition, by slowing down the precipitation runoff, the system elevates the resilience to floods and droughts in the area, which is of paramount importance in the face of climate change. The control algorithm does not assume the same price pattern or the scheduling horizon used by different stakeholders. The HP owners are not forced to co-operate—they only need to broadcast their control decisions to neighbors. As opposed to the adverse PV “inverter combat” [35], the co-operation among prosumers in the proposed scheme boosts their profits. Also, contrary to pumped HPs, energy accumulation does not incur extra costs, so even small price variations allow for increasing the revenue.

The presented algorithm performs better than classical optimization procedures in terms of numerical stability and robustness [29]. It can be implemented with negligible computing resources and triggered asynchronously at the HPs. While the final income depends on the weather, forecast accuracy, variability of energy prices, and other external conditions, the increase compared to a non-controlled case falls in the 20–70% range and should grow with the impact of RSEs, especially PV and wind RSEs.

Author Contributions: Conceptualization, M.M. and P.I.; methodology, M.M.; software, M.M.; validation, M.M. and P.I.; formal analysis, M.M. and P.I.; investigation, M.M.; resources, M.M. and P.I.; data curation, M.M. and P.I.; writing—original draft preparation, M.M.; writing—review and editing, P.I.; visualization, M.M. and P.I.; supervision, P.I.; project administration, P.I.; funding acquisition, P.I. All authors have read and agreed to the published version of the manuscript.

Funding: This work was performed in the framework of project no. 2021/41/B/ST7/00108: “Robust control solutions for multi-channel networked flows”, financed by the National Science Centre, Poland.

Data Availability Statement: All the data are available publicly [5].

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Kolosok, S.; Bilan; Vasylieva, T.; Wojciechowski, A.; Morawski, M. A scoping review of renewable energy, sustainability and the environment. *Energies* **2021**, *14*, 4490. [[CrossRef](#)]

2. Rahman, M.M.; Oni, A.O.; Gemechu, E.; Kumar, A. Assessment of energy storage technologies: A review. *Energy Convers. Manag.* **2020**, *223*, 113295. [CrossRef]
3. Wirfs-Brock, J. IE Questions: Why Is California Trying to Behead the Duck? 2014. Available online: <https://insideenergy.org/2014/10/02/ie-questions-why-is-california-trying-to-behead-the-duck/> (accessed on 12 May 2023).
4. Borkowski, D.; Cholewa, D.; Korzeń, A. A run-of-the-river Hydro-PV battery hybrid system as an energy supplier for local loads. *Energies* **2021**, *14*, 5160. [CrossRef]
5. PSE. Market Energy Prices. Available online: <https://www.pse.pl/dane-systemowe/funkcjonowanie-rb/raporty-dobowe-z-funkcjonowania-rb/podstawowe-wskazniki-cenowe-i-kosztowe/rynkowa-cena-energii-elektrycznej-rce> (accessed on 1 August 2023).
6. Anaza, S.O.; Abdulazeez, M.S.; Yisah, Y.A.; Yusuf, Y.O.; Salawu, B.U.; Momoh, S.U. Micro Hydro-Electric Energy Generation—An Overview. *Am. J. Eng. Res.* **2017**, *6*, 5–12.
7. Sachdev, H.; Akella, A.K.; Kumar, N. Analysis and evaluation of small hydropower plants: A bibliographical survey. *Renew. Sustain. Energy Rev.* **2015**, *51*, 1013–1022. [CrossRef]
8. Yiwei, Q.; Jin, L.; Liu, F.; Dai, N.; Song, Y.; Chen, G.; Ding, L. Tertiary Regulation of Cascaded Run-of-the-River Hydropower in the Islanded Renewable Power System Considering Multi-Timescale Dynamics. *IET Renew. Power Gener.* **2021**, *15*, 1778–1795.
9. Jamali, S.; Jamali, B. Cascade hydropower systems optimal operation: Implications for Iran’s Great Karun hydropower systems. *Appl. Water Sci.* **2019**, *9*, 66. [CrossRef]
10. Wyszowski, K.; Piwowarek, Z.; Pałejko, Z. Małe Elektrownie Wodne w Polsce. Technical Report, UN Global. 2022. Available online: https://ungc.org.pl/wp-content/uploads/2022/03/Raport_Male_elektrownie_wodne_w_Polsce.pdf (accessed on 1 June 2024). (In Polish).
11. YouTube. Constriction General Channel. 2024. Available online: https://www.youtube.com/watch?v=MBDV_Xj6Dj8 (accessed on 1 June 2024).
12. IEA Hydropower. Management Models for Hydropower Cascade Reservoirs, Compilation of Cases. Tech Report. 2021. Available online: https://www.ieahydro.org/media/02e6ec8e/EAHydro_AnnexXIV_Management%20Models%20for%20Hydropower%20Cascade%20Reservoirs_CASE%20COMPILATION.pdf (accessed on 1 June 2024).
13. Yu, Y.; Wu, Y.; Sheng, Q. Optimal scheduling strategy of cascade hydropower plants under the joint market of day-ahead energy and frequency regulation. *IEEE Access* **2021**, *9*, 87749–87762. [CrossRef]
14. Zhang, H.; Chang, J.; Gao, C.; Wu, H.; Wang, Y.; Lei, K.; Long, R.; Zhang, L. Cascade hydropower plants operation considering comprehensive ecological water demands. *Energy Convers. Manag.* **2019**, *180*, 119–133. [CrossRef]
15. Thaeer Hammid, A.; Awad, O.I.; Sulaiman, M.H.; Gunasekaran, S.S.; Mostafa, S.A.; Manoj Kumar, N.; Khalaf, B.A.; Al-Jawhar, Y.A.; Abdulhasan, R.A. A Review of Optimization Algorithms in Solving Hydro Generation Scheduling Problems. *Energies* **2020**, *13*, 2787. [CrossRef]
16. Mielczarski, W. *Energy Systems & Markets*; Association of Polish Electrical Engineers: Warszawa, Poland, 2018.
17. Singh, V.K.; Singal, S. Operation of hydro power plants—A review. *Renew. Sustain. Energy Rev.* **2017**, *69*, 610–619. [CrossRef]
18. Li, J.; Zhao, Z.; Li, P.; Mahmud, M.A.; Liu, Y.; Chen, D.; Han, W. Comprehensive benefit evaluations for integrating off-river pumped hydro storage and floating photovoltaic. *Energy Convers. Manag.* **2023**, *296*, 117651. [CrossRef]
19. Wijesinghe, A.; Lai, L. Small hydro power plant analysis and development. In Proceedings of the 2011 4th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT), Weihai, China, 6–9 July 2011; pp. 25–30.
20. Khomsah, A.; Laksono, A.S. Pico-hydro as A Renewable Energy: Local Natural Resources and Equipment Availability in Efforts to Generate Electricity. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *462*, 012047. [CrossRef]
21. Anderson, D.; Moggridge, H.; Warren, P.; Shucksmith, J. The impacts of ‘run-of-river’ hydropower on the physical and ecological condition of rivers. *Water Environ. J.* **2015**, *29*, 268–276. [CrossRef]
22. Nguyen, A.; Vu, V.; Hoang, D.; Nguyen, T.; Nguyen, K.; Nguyen, P.; Ji, Y. Attentional ensemble model for accurate discharge and water level prediction with training data enhancement. *Eng. Appl. Artif. Intell.* **2023**, *126*, 107073. [CrossRef]
23. Alvarez, G. An optimization model for operations of large scale hydro power plants. *IEEE Lat. Am. Trans.* **2020**, *18*, 1631–1638. [CrossRef]
24. Shaw, A.R.; Sawyer, H.; LeBoeuf, E.; McDonald, M.; Hadjerioua, B. Hydropower optimization using artificial neural network surrogate models of a high-fidelity hydrodynamics and water quality model. *Water Resour. Res.* **2017**, *53*, 9444–9461. [CrossRef]
25. Ahmadianfar, I.; Samadi-Koucheksaraee, A.; Asadzadeh, M. Extract nonlinear operating rules of multi-reservoir systems using an efficient optimization method. *Sci. Rep.* **2022**, *12*, 18880. [CrossRef]
26. Bernardes, J., Jr.; Santos, M.; Abreu, T.; Prado, L., Jr.; Miranda, D.; Julio, R.; Viana, P.; Fonseca, M.; Bortoni, E.; Bastos, G.S. Hydropower operation optimization using machine learning: A systematic review. *AI* **2022**, *3*, 78–99. [CrossRef]
27. Marcelino, C.G.; Leite, G.M.C.; Delgado, C.A.D.M.; de Oliveira, L.B.; Wanner, E.F.; Jiménez-Fernández, S.; Salcedo-Sanz, S. An efficient multi-objective evolutionary approach for solving the operation of multi-reservoir system scheduling in hydro-power plants. *Expert Syst. Appl.* **2021**, *185*, 115638. [CrossRef]
28. Linke, H. A model-predictive controller for optimal hydro-power utilization of river reservoirs. In Proceedings of the 2010 IEEE International Conference on Control Applications, Yokohama, Japan, 8–10 September 2010; pp. 1868–1873.
29. Morawski, M.; Ignaciuk, P. Balancing energy budget in prosumer water plant installations with explicit consideration of flow delay. *IFAC-PapersOnLine* **2023**, *56*, 11748–11753. [CrossRef]

30. Ignaciuk, P.; Morawski, M. Optimal Control of Cascade Hydro Plants as a Prosumer-Oriented Distributed Energy Depot. *Energies* **2024**, *17*, 469. [[CrossRef](#)]
31. Danielson, C.; Borrellia, F.; Oliver, D.; Anderson, D.; Phillips, T. Constrained flow control in storage networks: Capacity maximization and balancing. *Automatica* **2013**, *49*, 2612–2621. [[CrossRef](#)]
32. Basin, M.V.; Guerra-Avellaneda, F.; Shtessel, Y.B. Stock management problem: Adaptive fixed-time convergent continuous controller design. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 4974–4983. [[CrossRef](#)]
33. Ignaciuk, P. Dead-time compensation in continuous-review perishable inventory systems with multiple supply alternatives. *J. Process Control* **2012**, *22*, 915–924. [[CrossRef](#)]
34. Sengul, C. Message Queuing Telemetry Transport (MQTT) and Transport Layer Security (TLS) Profile of Authentication and Authorization for Constrained Environments (ACE) Framework; RFC 9431. 2023. Available online: <https://datatracker.ietf.org/doc/rfc9431/> (accessed on 1 June 2024).
35. Gu, Y.; Green, T.C. Power System Stability With a High Penetration of Inverter-Based Resources. *Proc. IEEE* **2023**, *111*, 832–853. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Energy-Efficient Neural Network Acceleration Using Most Significant Bit-Guided Approximate Multiplier

Pengfei Huang , Bin Gong, Ke Chen * and Chenghua Wang

College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, No. 29 Jiangjun Avenue, Jiangning District, Nanjing 210016, China; pffhuang@nuaa.edu.cn (P.H.); bingong@nuaa.edu.cn (B.G.); chwang@nuaa.edu.cn (C.W.)

* Correspondence: chen.ke@nuaa.edu.cn

Abstract: The escalating computational demands of deep learning and large-scale models have led to a significant increase in energy consumption, highlighting the urgent need for more energy-efficient hardware designs. This study presents a novel weight approximation strategy specifically designed for quantized neural networks (NNs), resulting in the development of an efficient approximate multiplier leveraging most significant one (MSO) shifting. Compared to both energy-efficient logarithmic approximate multipliers and accuracy-prioritized non-logarithmic approximate multipliers, our proposed logarithmic-like design achieves an unparalleled balance between accuracy and hardware costs. When compared with the baseline exact multiplier, our innovative design exhibits remarkable reductions, encompassing a decrease of up to 28.31% in area, a notable 57.84% reduction in power consumption, and a diminution of 11.86% in delay. Experimental outcomes reveal that the proposed multiplier, when applied in neural networks, can conserve approximately 60% of energy without compromising task accuracy. Concurrently, experiments focused on the transformer accelerator and image processing illustrate the substantial energy savings that can be obtained for Large Language Models (LLMs) and image processing tasks through the implementation of our proposed design, further validating its efficacy and practicality.

Keywords: approximate computing; large language model; coefficient multiplier; logarithmic multiplier



Citation: Huang, P.; Gong, B.; Chen, K.; Wang, C. Energy-Efficient Neural Network Acceleration Using Most Significant Bit-Guided Approximate Multiplier. *Electronics* **2024**, *13*, 3034. <https://doi.org/10.3390/electronics13153034>

Academic Editors: Ke Xie, Shoue Chen and Xiaolong Wang

Received: 5 July 2024

Revised: 30 July 2024

Accepted: 31 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The field of artificial intelligence has experienced significant progress, notably with the emergence of deep neural networks (DNNs) and Large Language Models (LLMs). However, these sophisticated models have introduced considerable computational and memory challenges, stemming from their ever-growing sizes, as thoroughly analyzed in [1]. These challenges are even more acute for edge devices, which are constrained by inherent limitations associated with their computational power. To enable the practical deployment of neural networks (NNs) on edge platforms, diverse software-based model compression techniques, such as quantization and pruning, have been presented as feasible solutions. Additionally, recent efforts leveraging the approximate computing (AxC) paradigm at the circuit level have persuasively demonstrated its capacity to improve system performance [2]. This improvement results from the reduction in computational complexity and memory demands, attained through replacing precise arithmetic elements with their less accurate equivalents. The key operations of arithmetic processors—namely, multiplication and addition—significantly impact their overall energy consumption. Notably, multiplication consumes more energy than addition. The core arithmetic computation performed within NNs is the multiply–accumulate (MAC) operation. As millions of multiplication operations are performed by an NN accelerator, approximate multipliers (AMs) have garnered extensive research attention due to their considerable significance in terms of improving the efficiency and accuracy of neural network applications [3–5].

Previous research on approximate multipliers customized for DNNs has delivered encouraging results regarding efficiency [6]. In the case of deploying NNs on resource-constrained edge devices, the employment of approximate multipliers provides an appealing means to strike a balance between accuracy and energy cost. This accomplishment relies on alleviating and relaxing the stringent requirements for perfectly precise multiplication outcomes, thus facilitating the creation of approximate hardware designs that are more resource- and energy-efficient. The study in [7] demonstrated the efficiency of AxC in DNNs through presenting a new AxC DNN layer that incorporates approximate multipliers into the DNN design process. In [8], three hybrid approximate multipliers were proposed by employing NAND gates instead of AND gates and applying different combinations of the compressors to provide diverse trade-offs between accuracy and hardware efficiency. An open-source adaptable approximate multiplier design, driven by input distribution and polarity, was suggested in [9] to generate optimized approximate multipliers. The purpose is to minimize the average square of the absolute error of an approximate multiplier based on the probability distributions of operands extracted from the target application, with the input polarity taken into consideration. In [10], a methodology for designing approximate array multipliers leveraging the concept of carry disregard was introduced. Through strategically selecting the positions for carry disregard, it becomes feasible to disregard a small number of carries, thereby achieving enhanced speed and a reduced area while simultaneously maintaining superior overall accuracy and ensuring suitability for the intended application.

Approximate logarithmic multipliers provide a simpler design but lead to a notably higher accuracy loss, whereas approximate non-logarithmic multipliers demonstrate a lower computational error at the cost of an increased design complexity [11]. To address the need for reduced energy consumption while maintaining an acceptable accuracy loss, this study proposes an energy-efficient most significant one-driven shifting multiplier (MSAM). The MSAM can obtain a range of highly efficient multiplier designs by dynamically configuring the approximation factor (k) and the precision factor (m). By leveraging the approximation factor (k), the multiplier is strategically divided into exact and approximate sub-multipliers. Within the approximation region, the re-configurable constant coefficient multiplier (RCCM) principle [12] is applied, substituting conventional multiplication operations with shift, addition, and subtraction operations. This journal article builds upon and extends our previous conference paper [13], offering several key differences and novel contributions. A summary of these advancements is provided below:

- We devised an alternative and more aggressive approximation strategy for the multiplier approximation process, pushing the boundaries of accuracy–efficiency trade-offs.
- We further conducted a comprehensive error analysis and thoroughly discuss the trade-offs between the accuracy and energy consumption.
- In addition to analyzing popular data sets, we also explore image processing using different AxC schemas.
- The advantages of our proposed approximate multiplier designs are substantiated through a rigorous theoretical analysis and extensive experimental demonstrations.

The remainder of this paper is structured as follows: In Section 2, a concise review of the background and related work is offered. Section 3 details our innovative methodology for energy-efficient neural network acceleration, incorporating a most significant bit-guided dynamic approximate multiplier to significantly enhance performance. In Section 4, an error analysis and simulation results are provided. Finally, Section 5 concludes the paper.

2. Preliminaries

A logarithmic multiplier (LM) converts numbers in binary format into logarithmic format, facilitating the transformation from multiplication to addition. Mitchell's algorithm was initially proposed in [14]. Given two binary numbers A and B , where $A = 2^{k_1}(1 + x_1)$

and $B = 2^{k_2}(1 + x_2)$, the logarithmic multiplier initially converts them into their logarithmic forms as follows:

$$\log_2(A \cdot B) = \log_2(2^{k_1+k_2}(1 + x_1)(1 + x_2)) = (k_1 + k_2) + \log_2(1 + x_1) + \log_2(1 + x_2), \quad (1)$$

where k_1 and k_2 denote the positions of the most significant bit (with a value of 1) in operands A and B , respectively, and x_1 and x_2 represent the fractional parts of A and B ($x_1, x_2 \in [0, 1]$). In Mitchell's algorithm, when $0 \leq x < 1$, the expression $\log_2(1 + x)$ is approximately equal to x . Therefore, the logarithm of the product can be expressed as follows:

$$\log_2(A \cdot B) \approx k_1 + k_2 + x_1 + x_2. \quad (2)$$

The logarithmic multiplier encompasses several crucial components: leading-one detectors (LODs), logarithmic converters, an adder, and an anti-logarithmic converter. The LODs are tasked with identifying the most significant 1 bit within the input binary numbers. Subsequently, the logarithmic converters generate the logarithmic representations of these binary numbers. These logarithmic values are added together to obtain the product, which is then decoded back into a binary number using the anti-logarithmic converter.

3. The Proposed Most Significant One-Driven Shifting Approximate Multiplier

In this study, we introduce a most significant one-driven shifting approximate multiplier that employs a logarithmic-like approach. This novel multiplier achieves improved accuracy while maintaining hardware costs comparable to, or even lower than, those of traditional approximate logarithmic multipliers.

3.1. The Architecture of the Design

The fundamental operations in neural networks (NNs) involve multiplications, specifically the multiplication of weights w by activations x . The architecture for an NN utilizing the proposed most significant one-driven shifting approximate multiplier (MSAM) is illustrated in Figure 1. The MSAM primarily comprises two core components: an approximate sub-multiplier (ASM) and an exact sub-multiplier (ESM), enhanced with auxiliary shifters and adders for a straightforward and coherent structural design. Specifically, the weight w is divided into two parts based on the significance of its bits: the most significant bits form w_H , processed by the ESM, while the less significant bits are manipulated to form w'_L , processed by the ASM. Subsequently, the result from the ESM is shifted, and an addition operation is performed to generate the final result r .

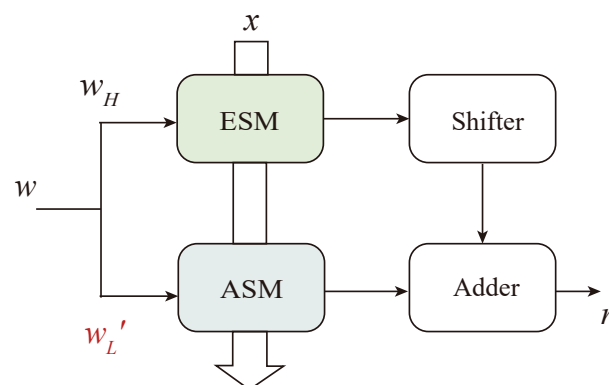


Figure 1. The architecture of the most significant one-driven shifting approximate multiplier (MSAM) block design.

A sign converter similar to that in [15] is employed to extend the MSAM to support signed numbers, as illustrated in the left part of Figure 2. The internal feature of the MSAM block is also presented in Figure 2. We introduce two pivotal parameter factors that confer the attribute of dynamic configurability to the multiplier, catering to a diverse

range of precision requirements that might fluctuate across various application contexts. The first is the approximation factor, denoted as k , which is utilized for the partitioning of the multiplier. This partitioning operation separates the n -bit multiplier into two distinct segments: k lower-order bits and $(n - k)$ higher-order bits. This partitioning scheme effectively decomposes a solitary multiplication operation into two discrete multiplicative processes involving the multiplicand and two multipliers with smaller bit widths. Second, the precision factor, denoted as m , is incorporated within the approximation strategy governing the ASM, and plays a crucial role in facilitating configurable precision settings within the scope of the sub-multiplier.

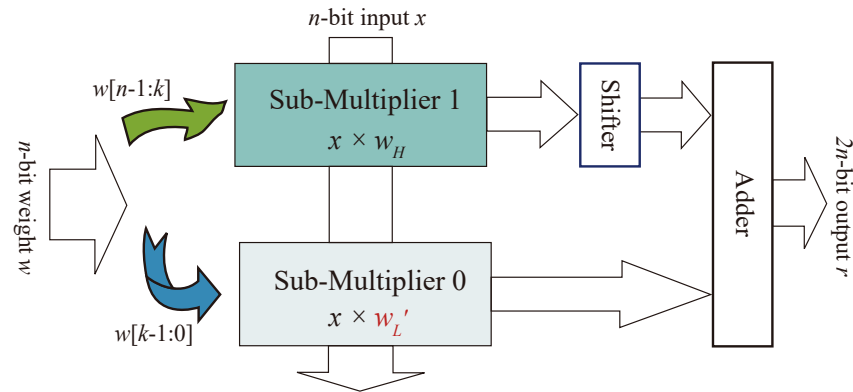


Figure 2. Extension of the proposed approach to support signed numbers, including internal feature of MSAM block.

3.2. The Proposed Dynamic Weight Approximation Strategies

To achieve a balance between computational efficiency and accuracy in the MSAM design, we introduce multiple weight approximation strategies tailored for the ASM component, based on the fundamental principles of the logarithmic multiplier (LM) and re-configurable constant coefficient multiplier (RCCM) design. These strategies aim to accommodate various application requirements and optimize performance under different conditions, encompassing the transformation of multiplication operations into simpler arithmetic operations such as shifts, addition, or subtraction. Meanwhile, the ESM component follows conventional, precise multiplication techniques to ensure accurate computation. To ensure accurate cumulative addition, the output generated by the ESM undergoes a sequence of shifting operations to align it with the output from the ASM. This alignment is crucial for the cumulative addition process, which ultimately determines the final output of the MSAM block. By combining these distinct strategies in the ASM and ESM components, we aim to optimize the overall performance of the MSAM design.

The activation value x serves as the multiplicand, while the weight w assumes the role of the multiplier in the multiplication operation within a neural network (NN). To optimize the approximate sub-multiplier (ASM) for neural networks, we propose a method that partitions the model's weights into distinct data ranges and approximates the weights within these ranges as constant values. This collection of approximate constant values, denoted as W' , effectively corresponds to the constant coefficient set C employed in the RCCM framework [12]. A fundamental difference is that the proposed set W' requires the determination of only a sole initial constant, w'_0 , while the remaining constants in the set are deduced with respect to this initial constant. Furthermore, this innovative weight approximation strategy enables the dynamic selection of appropriate approximate constant values based on the most significant one (MSO) in the input weight data.

3.2.1. One-Dominating Approximation Strategy (OAS)

The process of single-multiplier (w) approximation using the proposed method is depicted in Figure 3. Leading-one detection (LOD) is utilized to detect the most significant one (MSO) of the multiplier w . Once the MSO is identified, the subsequent $(m - 1)$ bits

are set to 1, while the remaining bits are set to 0. If the position of the MSO is at the mso -th bit (counting from 0, where mso is the index of the MSO bit), all numbers within the range of $w \in [2^{mso}, 2^{mso+1} - 1]$ are approximated as a constant value of one, according to Equation (3).

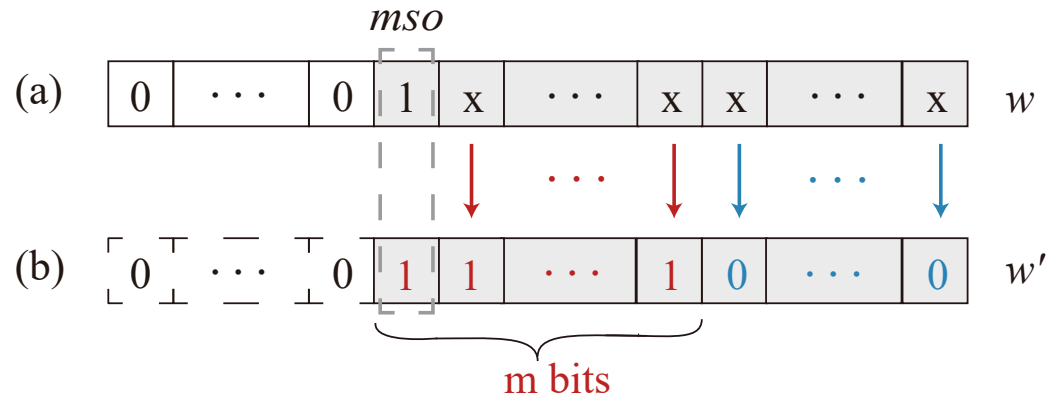


Figure 3. Example of the multiplier one-dominating approximation process: (a) original number; (b) approximated number.

$$w' = \begin{cases} w'_0, & mso < m \\ \sum_{i=mso-m+1}^{mso} 2^i, & mso \geq m. \end{cases} \quad (3)$$

The value of w'_0 is determined by the factor m . For instance, when $m = 1$, w'_0 is set to $1b$; meanwhile, when $m = 2$, w'_0 is set to $11b$. In this approximation method, the multiplier is dynamically split into separate data range intervals based on the position of the MSO in the input data. These intervals are then approximated using a set of constant coefficients. We refer to m as the precision factor, as it controls the number of approximation intervals.

The selection of m consecutive bits starting from the MSO and setting them to 1 is based on two principal rationales. First, this enables the establishment of a connection between this particular group of approximate values and the initial approximate constant coefficient w'_0 through a succession of shifting operations. As a result, through offering an initial approximate constant w'_0 and relying on the product of x and w'_0 , we can effectively reconstitute the multiplication outputs of the remaining coefficients in relation to the input x . This method significantly streamlines the hardware implementation and lowers complexity. Second, a sensible choice of the value of m allows for the strategic positioning of the approximate constants near the average value within each weight interval, thus maximizing the precision of the multiplier. In essence, it allows the approximated constants to gravitate toward the central value of the individual weight range, thereby enhancing the overall accuracy of the multiplier architecture.

3.2.2. Zero-Dominating Approximation Strategy (ZAS)

In the above one-dominating approximation strategy, all m bits starting from the MSO position are set to 1 for an important reason: this approximated m -bit number should closely resemble the average value of all k -bit numbers. However, in the extreme case where $m = k$, the approximated number becomes the largest, rather than approximating the mean value. To resolve this dilemma, we propose an alternative zero-dominating approximation strategy, as illustrated in Figure 4. Immediately after the position of the MSO, not all the remaining bits are set to 1. Instead, the first bit immediately following the MSO position is set to 0, with the rest of those bits being set to 1. This effectively avoids the issue of excessively large approximate numbers while ensuring proximity to the mean value. With a 0 acting as a separator, this also allows for reasonable application of carry discard techniques [10], thereby reducing latency and power consumption.

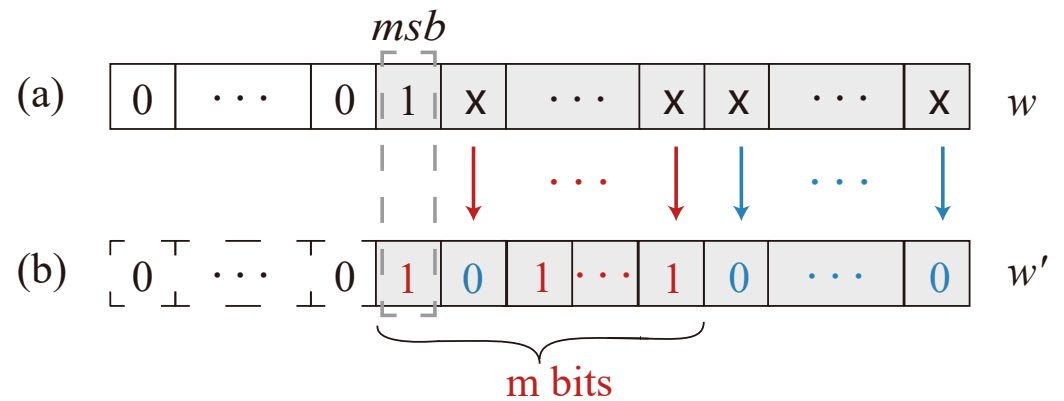


Figure 4. Example of the multiplier zero-dominating approximation process: (a) original number and (b) approximated number.

3.2.3. Compensation for Zero-Dominating Approximation Strategy

To better leverage the advantages of approximation, we typically do not choose $m = k$. Empirically, setting $m \leq \log_2(k) + 1$ tends to provide a favorable balance between energy consumption and accuracy benefits.

As shown in Table 1, in terms of the MSAM-42 approach (i.e., $k = 4$ and $m = 2$), the average result obtained with the OAS is slightly larger than the median, while the average result with the ZAS is significantly smaller than the median.

Table 1. Results obtained with MSAM-42¹ using various approximation strategies.

The k Bits (Input $w[3:0]$)	One-Dominating Approximation	Zero-Dominating Approximation	Compensation for Zero-Dominating
0000	0000	0000	0000
0001 0010 0011	0011	0010	0101
0100 0101 0110 0111	0110	0100	0101
1001 1010 1011 1100 1101 1110 1111	1100	1000	1010

¹ MSAM-42 indicates that the approximate factor k is 4 and the corresponding precision factor m is 2.

Although this can better curtail energy consumption, it may also bring excessive errors. To achieve a better equilibrium in this regard, we introduce a compensation scheme to mitigate errors and guarantee that the overall accuracy is not notably compromised. As depicted in Figure 5, considering that the approximated number with ZOS is likely smaller than the actual one, through adding an extra compensation of 1 after the m -bit, we can effectively reduce the error distance while the additional energy consumption is essentially negligible.

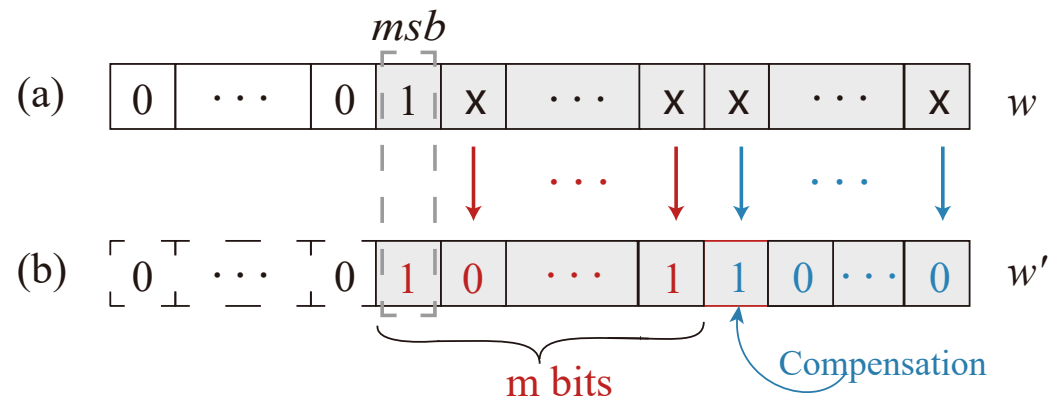


Figure 5. Example of the multiplier zero-dominating approximation with compensation process: (a) original number and (b) approximated number.

3.3. Implementation of the Hardware Design

The core of the MSAM design's excellence lies in its attentive management of the ASM, as refining its hardware directly and significantly affects the overall energy consumption of the multiplication unit.

As illustrated in Figure 6, the process commences with a series of shift and addition operations, jointly generating the product of the multiplicand x and the specific substitution value w'_0 for w_L in accordance with the factors k and m . The position (pos) of the most significant one bit within the input weight w_L , identified by the leading-one detector (LOD), establishes the control signal for the subsequent multiplexer (MUX). This signal governs the shift in $r0_{tmp}$. Subsequently, the multiplexer is utilized to generate the output $r0$ for the approximate sub-multiplier (ASM) module. Meanwhile, the ESM features an accurate approach to multiplication. Eventually, the output from the ESM undergoes a succession of shifting operations and is then merged with the outcome from the ASM, ultimately yielding the final output of the MSAM.

To accommodate signed operations, the MSAM architecture can be augmented through integrating sign conversion circuits both before and after unsigned operations, as depicted in Figure 2. These circuits employ the complement and execute an OR operation with a logic 1b in the least significant bit, serving as an approximation for negation when dealing with negative inputs.

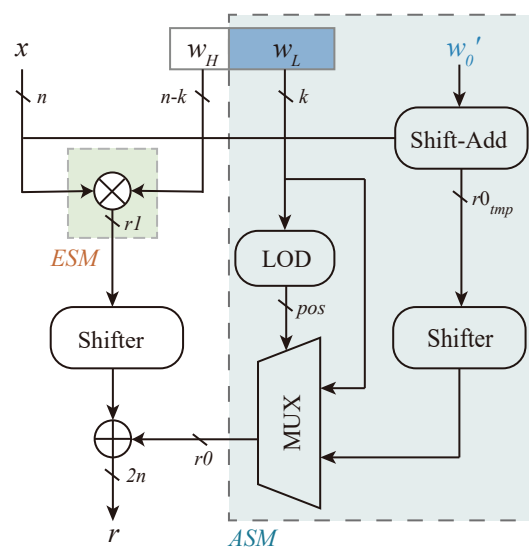


Figure 6. Hardware design implementation of MSAM.

4. Results and Discussion

Our proposed MSAM design is implemented using Verilog HDL, along with other advanced approximate multipliers. The designs were synthesized using the Synopsys Design Compiler (DC) with 28 nm CMOS technology, and the simulation was conducted using approximately 1,000,000 sets of random numbers, covering the entire input space of 8×8 multipliers (65,536 combinations) at a clock frequency of 200 MHz. The VCS tool was utilized for the simulation, generating back-annotated switching activity files to facilitate power analysis. The NMED metric was employed to quantify the error of the proposed designs, which are comprehensively evaluated across the entire input space using uniformly distributed data. The proposed unsigned MSAMs were compared with Mitch-w [14], DR-ALM-7 [15], DRUM6 [16], and AxRM2 [17], while the signed MSAMs were compared with Mitch-w [14], DR-ALM-7 [15], C-SSM ($m = 4$) [18], and Approx-Booth [19]. For clear and convenient comparisons, MSAM refers to the use of the OSA approximation strategy, MSAM^Z indicates the use of the ZSA approximation strategy, and MSAM^C denotes ZSA with additional compensation.

The abbreviations used in this manuscript are presented in Table 2.

Table 2. Abbreviations for approximate multiplier designs.

Multiplier	Techniques	Description
Exact	-	Exact multiplier.
MSAM- km	Logarithmic-like	The proposed most significant one-driven shifting approximate multiplier under varying approximate factors k and precision factors m .
Mitch-w [14]	Logarithmic	Mitchell's unsigned approximate logarithmic multiplier with w -bit operand.
Mitch-w* [14]	Logarithmic	Mitchell's signed approximate logarithmic multiplier with w -bit operand.
DR-ALM-7 [15]	Logarithmic	Approximate non-iterative unsigned logarithmic multiplier with L_i -bit operand ($L_i = 7$).
DR-ALM-7* [15]	Logarithmic	Approximate non-iterative signed logarithmic multiplier with L_i -bit operand ($L_i = 7$).
DRUM6 [16]	Logarithmic	Unbiased multiplier with $k(k = 6)$ bits in the operate from the leading one.
AxRM2 [17]	Non-logarithmic	Approximate recursive multipliers using high-performance building blocks.
C-SSM ($m = 4$) [18]	Non-logarithmic	Approximate static segmented multiplier with a contiguous segment of m bits ($m = 4$).
Approx-Booth [19]	Non-logarithmic	Low-cost approximate radix-4 Booth multiplier with simplified compressor and encoder designs.

4.1. Error Analysis

Deep learning accelerators that incorporate approximate multipliers and adders have been devised to mitigate energy consumption and delay, albeit with an acceptable trade-off in accuracy [20]. As a general principle, a higher degree of acceleration tends to result in a greater reduction in accuracy. The error characteristics of an approximate multiplier (AM) are typically ascertained through exhaustive simulations that encompass all conceivable input combinations or, alternatively, through Monte Carlo simulations that utilize random inputs sampled from a pre-determined distribution, such as the normal or uniform distribution.

Consider that $\mathcal{M}^*$ denotes the output of the approximate multiplier and $\mathcal{M}$ represents the exact output. The error distance (ED) provides a precise measurement of the absolute difference between these two outputs, which is formulated as $ED = |\mathcal{M} - \mathcal{M}^*|$. The relative error distance (RED) expresses the error as a percentage relative to the exact result. Furthermore, the mean error distance (MED) and the mean relative error distance (MRED) are widely used metrics for evaluating the accuracy of approximate designs. In addition

to MED and MRED, the normalized MED (NMED) and normalized RED (NRED) are stable metrics that are independent of the implementation size, making them valuable for assessing approximate designs of varying sizes. As demonstrated in [21], the NMED of an approximate multiplier (AM) has a significant impact on the output error of a neuron in a neural network, particularly when dealing with a large number of inputs.

For the specified MSAM, with the value of k remaining constant and ensuring the consistency of the operand bit-width for all operations, we employ MRED and NMED, as two effective and widely used metrics, in order to evaluate the error introduced by the ASM block. These metrics serve as indispensable means for the assessment of the errors inherent in the proposed design, with a focus on different configurations of the precision factor m . Figure 7 presents the MRED curves for k values of 3, 4, and 5, plotted against variations in the precision factor m , enabling the identification of distinct trends within the curves. For different values of k , the MRED reaches its minimum when m is 1 or 2. Contrary to other AMs, in the design of the MSAM, the precision factor m is not necessarily better when it is larger. In fact, when m exceeds $\log_2 k + 1$, an excessive number of bits are set to 1, leading to a significant deviation in the approximated value from the actual value. Additionally, a larger m results in more bits participating in the calculation, thereby consuming more energy. It is worth noting that the results obtained from the NMED metric closely align with these experimental findings. Consequently, in subsequent comparative analysis and detailed examinations, we prioritized m values of 1 or 2. This strategic choice was made with the aim of attaining an optimum balance in the design, effectively reconciling the precision requirements and overhead hardware cost considerations.

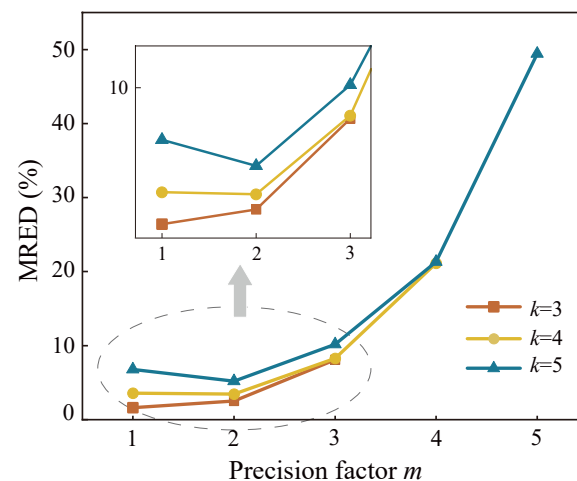


Figure 7. Mean relative error distance curves of MSAM under varying approximate factors k and precision factors m .

4.2. Synthesis and Simulation Results

Diverse MSAM- km multipliers, existing logarithmic multipliers, and advanced non-logarithmic multipliers were evaluated to assess their hardware performance and accuracy metrics.

In our approximate multiplier design, one operand is divided into two components: ESM and ASM, controlled by the approximation factor k , which determines the number of bits approximated. The precision factor m represents the degree of approximation. The level of approximation is primarily governed by k ; empirically, a larger k leads to a higher degree of approximation, simpler design, and lower energy consumption, while a smaller k results in less approximation and reduced energy savings. The parameter m decides how the k approximate bits are handled. A larger m retains more bits, but it is important to note that this does not mean retaining the original precise bits; instead, it uses m bits of all ones. Therefore, a larger m does not necessarily translate to higher precision. Statistically, an m closer to k tends to make k approximate bits all ones, leading to less precision. From

experimental results and statistical analysis, a reasonable value for m is around $\log_2(k)$, making m closer to the mean of all k -bit numbers. The introduction of parameters k and m enables the fine-tuning of the design to achieve the desired balance between accuracy and energy efficiency. However, these additional parameters might require meticulous consideration and optimization during the design phase to ensure optimal performance.

Logarithmic multipliers offer lower energy consumption but sacrifice accuracy. Our design mitigates this by providing a higher accuracy level while maintaining a lower power consumption than non-logarithmic multipliers. A comparison of NMED and the power delay product (PDP) for unsigned 8-bit multipliers is shown in Figure 8, where the proposed MSAMs are highlighted with a red square. While DR-ALM-7 [15] and Mitchell [14] exhibited lower PDP, they resulted in higher NMED. Conversely, DRUM6 [16] and MSAM-21 produced smaller NMED but demanded higher PDP. Notably, AxRM2 [17], MSAM-52, MSAM-41, and MSAM-42 achieved relatively lower PDP as well as smaller NMED. Among these, as indicated by the red circle, MSAM-41 and MSAM-42 emerge as the most appealing designs, achieving the optimal trade-off between PDP and NMED.

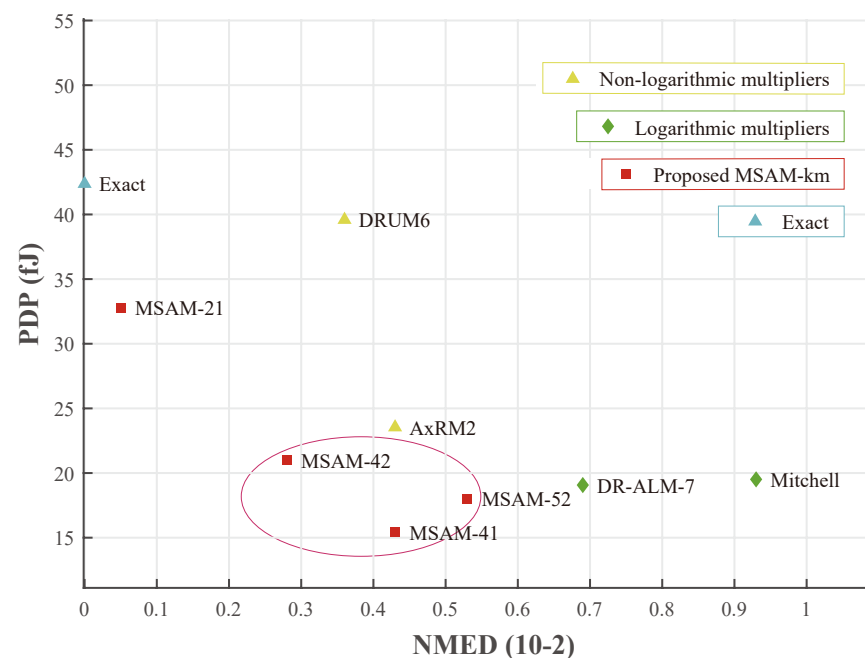


Figure 8. NMEDs and PDPs of 8-bit unsigned approximate multipliers.

The detailed results of this evaluation are summarized in Table 3. Through carefully configuring the parameters k and m , we attained a hardware performance comparable to (or, in some cases, exceeding) that of the Mitchell [14] and DR-ALM-7 [15] multipliers (for instance, with $km = 41$ or $km = 52$). When comparing MSAM-41 and MSAM-52 with DR-ALM-7, even in situations where the power delay product (PDP) performance surpasses the latter, remarkable improvements were still witnessed in terms of NMED, reaching 37.68% and 23.19%, respectively. In contrast, when compared to AxRM2 [17], MSAM-41 demonstrates significant enhancements, including a 32.11% reduction in area utilization, a 23.33% decrease in delay, and a 27.95% improvement in PDP, all while maintaining an equivalent NMED. These findings highlight the impressive hardware efficiency achieved with the MSAM design, when compared to state-of-the-art AM designs.

Although the ZAS offers a more simplified hardware design, resulting in superior energy consumption and area utilization, the NMED is increased compared to the corresponding OAS strategy. Our design provides multiple approximate strategies such as ZAS and OAS to achieve a more dynamic range of trade-offs, thereby meeting the demands of a wider range of applications.

Table 3. Performance comparison of 8-bit unsigned approximate multipliers.

Design	Area (um ²)	Power (uW)	Delay (ns)	PDP (fJ)	NMED (10 ^{−2})
Exact	154.48	76.75	0.59	45.28	0
MSAM-21	147.29	59.55	0.59	35.13 (22%)	0.05
MSAM ^Z -21	144.22	56.2	0.58	32.60 (28%)	0.07
MSAM-32	143.01	52.3	0.52	27.20 (40%)	0.17
MSAM ^Z -32	139.05	50.4	0.5	25.20 (44%)	0.19
MSAM-31	129.53	48.39	0.52	25.16 (44%)	0.17
MSAM ^Z -31	126.45	44.66	0.49	21.88 (52%)	0.19
MSAM-42	125.5	41.78	0.52	21.73 (52%)	0.28
MSAM ^Z -42	120.79	38.98	0.49	19.10 (58%)	0.34
MSAM-41	111.89	36.86	0.46	16.96 (63%)	0.43
MSAM ^Z -41	103.76	35.33	0.45	15.90 (65%)	0.47
MSAM-52	110.75	32.36	0.52	16.83 (63%)	0.53
MSAM ^Z -52	104.39	31.33	0.51	15.98 (65%)	0.59
Mitchell [14]	103.45	30.01	0.65	19.51 (−57%)	0.93
DR-ALM-7 [15]	98.28	28.46	0.67	19.07 (−58%)	0.69
DRUM6 [16]	192.53	56.10	0.77	43.20 (−5%)	0.36
AxRM2 [17]	164.81	39.23	0.60	23.54 (−48%)	0.43

The proposed designs are in the shaded section.

4.3. Evaluation, Simulation, and Analysis

4.3.1. Evaluation of Performance on Deep Neural Networks

For a more comprehensive evaluation of the multiplier’s performance when used in large models, we conducted experiments using the CIFAR-10 data set and employed three representative deep neural networks: VGG-19, known for its deep convolutional layers; ResNet-50, which effectively mitigates the vanishing gradient problem in deep networks through the use of residual connections; and DenseNet-121, which achieves efficient feature utilization through dense connectivity mechanisms. We conducted experiments using these complex networks, aiming to thoroughly assess the multiplier’s performance and applicability. Additionally, we expanded our experimental analysis to the more challenging ImageNet data set through incorporating SqueezeNet, a compact yet efficient deep neural network architecture designed for optimized performance on constrained devices.

In order to assess the performance of the approximate multipliers, we employed the Adapt framework [22] to facilitate the simulation and evaluation of various multiplier designs. As shown in Figure 9, the inputs consist of the pretrained NN model, quantization configurations, and the design file of the approximate multiplier (based on Verilog HDL). Initially, the layer graph is constructed using the parameters of the convolutional layer and the adjacent batch normalization layer. Subsequently, the design file of the approximate multiplier was analyzed. Based on the quantization configurations, the corresponding approximate layers are generated. These approximate layers then replace the layers in the original layer graph. Following this, the reconstructed layer graph is processed through the AdaPT framework for retraining and inference, adhering to the specified retraining strategy. It is noteworthy that for the training data set, only a representative subset is required, which can be approximately 10% of the original training set, for calibration purposes. Lastly, performance reports are provided, detailing the accuracy and energy consumption.

Model retraining and fine-tuning can be employed to alleviate the accuracy deterioration arising from certain approximations, such as pruning and approximate multiplication [23]. These approaches enable the model to adapt and become more resistant to the errors introduced by such approximations. During the inference phase, these techniques facilitate the AxC models to incur lower hardware cost while attaining significantly higher accuracy. To evaluate the performance of the approximate multipliers effectively, experiments were carried out on the CIFAR-10 data set through direct inference without any

retraining. This methodology permitted a rapid assessment of the multipliers' performance. Meanwhile, for the ImageNet data set, a more efficient retraining strategy was adopted by selecting only 10% of the training subset. This strategy was aimed at performing model tuning during the training phase, enabling accuracy improvement and substantial energy reduction during the actual inference phase. It showcases the efficacy of retraining and model tuning in mitigating errors and enhancing accuracy without the encumbrance of expensive hardware requirements.

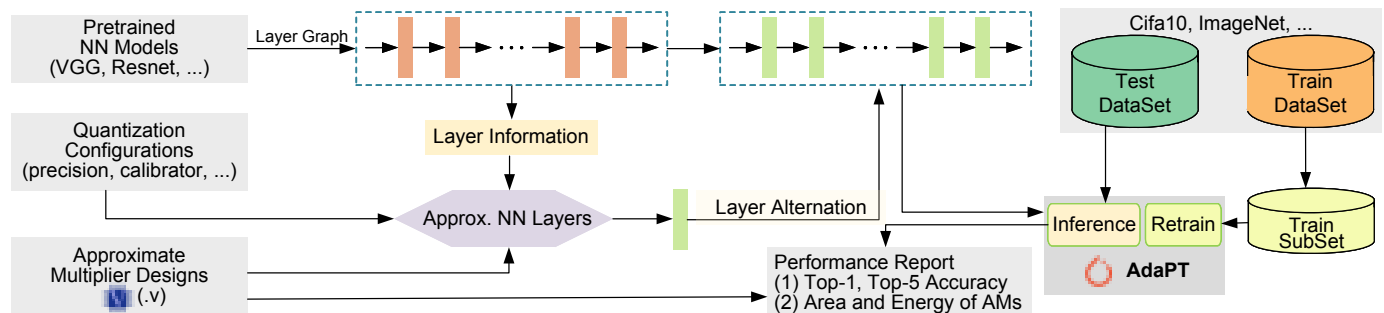


Figure 9. Evaluation flow based on AdaPT framework.

Several signed versions of the MSAM multiplier were implemented and thoroughly compared with the precise radix-4 Booth multiplier, alongside two logarithmic multipliers—namely, Mitchell [14] and DR-ALM-7 [15]—as well as two non-logarithmic approximate multipliers—C-SSM [18] and Approx-Booth [19]. To ensure a fair comparison, we replicated the designs of the 8-bit C-SSM (with $m = 4$) and Approx-Booth (with $w_G, w_A = 6, 6$). The evaluation of these signed multipliers strictly followed the previously outlined experimental setup. The results of this comprehensive comparison are presented in Table 4.

When considering the results on the CIFAR-10 data set, the MSAM approximate multiplier presented the best performance in terms of both accuracy and energy reduction. Specifically, the MSAM-31 multiplier not only outperformed the DR-ALM-7 in terms of network evaluation accuracy, but also achieved an additional 9% energy savings per single multiplier compared to DR-ALM-7, highlighting its efficiency. DR-ALM-7, on the other hand, demonstrated superior network performance when compared to the other approximate multipliers, with the exception of the MSAM variants.

Table 4. Comparison of the hardware costs and accuracies of 8-bit signed multipliers on the CIFAR10 and ImageNet data sets.

Multiplier	NMED (10^{-2})	Area (μm^2)	Power (μW)	Delay (ns)	Energy (fJ)	CIFAR10			ImageNet	
						VGG-19	ResNet-50	DenseNet-121	Top-1	Top-5
Exact radix-4	0	276.70	94.27	0.60	56.56	93.69%	93.57%	93.84%	57.37%	80.19%
Signed MSAM-21	0.34	173.88	52.82	0.62	32.75 (−42%)	93.66%	92.81%	93.58%	54.96%	78.21%
Signed MSAM-32	0.40	169.60	47.78	0.58	27.71 (−51%)	93.23%	91.34%	93.30%	52.02%	76.00%
Signed MSAM-31	0.61	156.11	42.82	0.55	23.55 (−58%)	93.60%	93.06%	93.64%	54.01%	77.60%
Signed MSAM-42	0.60	152.08	39.71	0.60	23.83 (−58%)	93.08%	91.71%	92.93%	50.37%	74.35%
Signed MSAM-41	1.20	138.47	34.10	0.49	16.71 (−70%)	91.24%	91.26%	91.46%	43.31%	68.00%
Mitchell* [14]	1.12	132.30	39.37	0.71	27.95 (−51%)	89.64%	92.52%	91.75%	50.19%	74.20%
DR-ALM-7* [15]	0.78	127.13	39.23	0.73	28.64 (−49%)	93.29%	92.24%	93.37%	53.92%	77.38%
C-SSM ($m = 4$) [18]	3.62	276.32	82.37	0.43	35.42 (−37%)	40.24%	61.17%	73.28%	2.22%	7.47%
Approx-Booth [19]	0.17	218.23	73.18	0.55	40.25 (−29%)	12.29%	10.65%	9.99%	17.44%	36.33%

The top three approximate multipliers that demonstrated the greatest adaptability to the ImageNet data set were MSAM-21, MSAM-31, and DR-ALM-7. These multipliers exhibited exceptional performance across the various network architectures, highlighting their versatility and robustness. For applications that prioritize high network accuracy,

MSAM-21 stood out as a notable performer, which achieved a 1.04% improvement in Top-1 accuracy and a 0.83% improvement in Top-5 accuracy, when compared to DR-ALM-7. These improvements are significant and can have a substantial impact on the overall performance of the network. The results presented in Table 4 provide a comprehensive comparison of the hardware costs and accuracies of the 8-bit signed multipliers on the CIFAR10 and ImageNet data sets. The MSAM and DR-ALM-7 multipliers consistently demonstrated strong performance, with the MSAM variants showing particular promise in terms of accuracy and energy efficiency. These findings have important implications for the design and optimization of multipliers for use in deep learning applications.

4.3.2. Performance Evaluation of Transformer Matrix Multiplication Accelerator

The resource and energy savings achieved with the proposed multiplier when deployed in LLMs on edge devices were verified. This verification was accomplished through the application of the multiplier to a transformer matrix multiplication accelerator, specifically designed to handle matrices of size 16×16 . When evaluating the hardware overhead, with exact multipliers serving as the baseline for comparison, we focused on assessing the power consumption of the accelerator associated with combinational logic. Ref. [24] highlights the possibility of performing inference in LLMs with up to 175B parameters without any performance degradation utilizing LLM.int8(), which enables the multiplication of over 99.9% of values in 8-bit. Ref. [25] demonstrates the efficiency of approximate multipliers in the context of vision transformer models. As depicted in Figure 10, our matrix multiplication accelerator involves two 8-bit integer matrices, X_{18} and W_{18} . To accelerate the multiplication process, we substituted the exact multiplication with an approximated method.

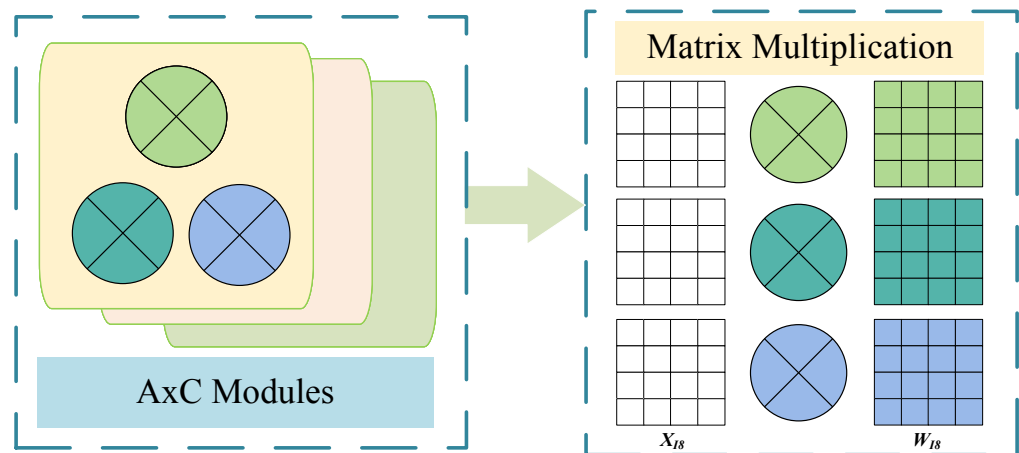


Figure 10. Matrix multiplication accelerator.

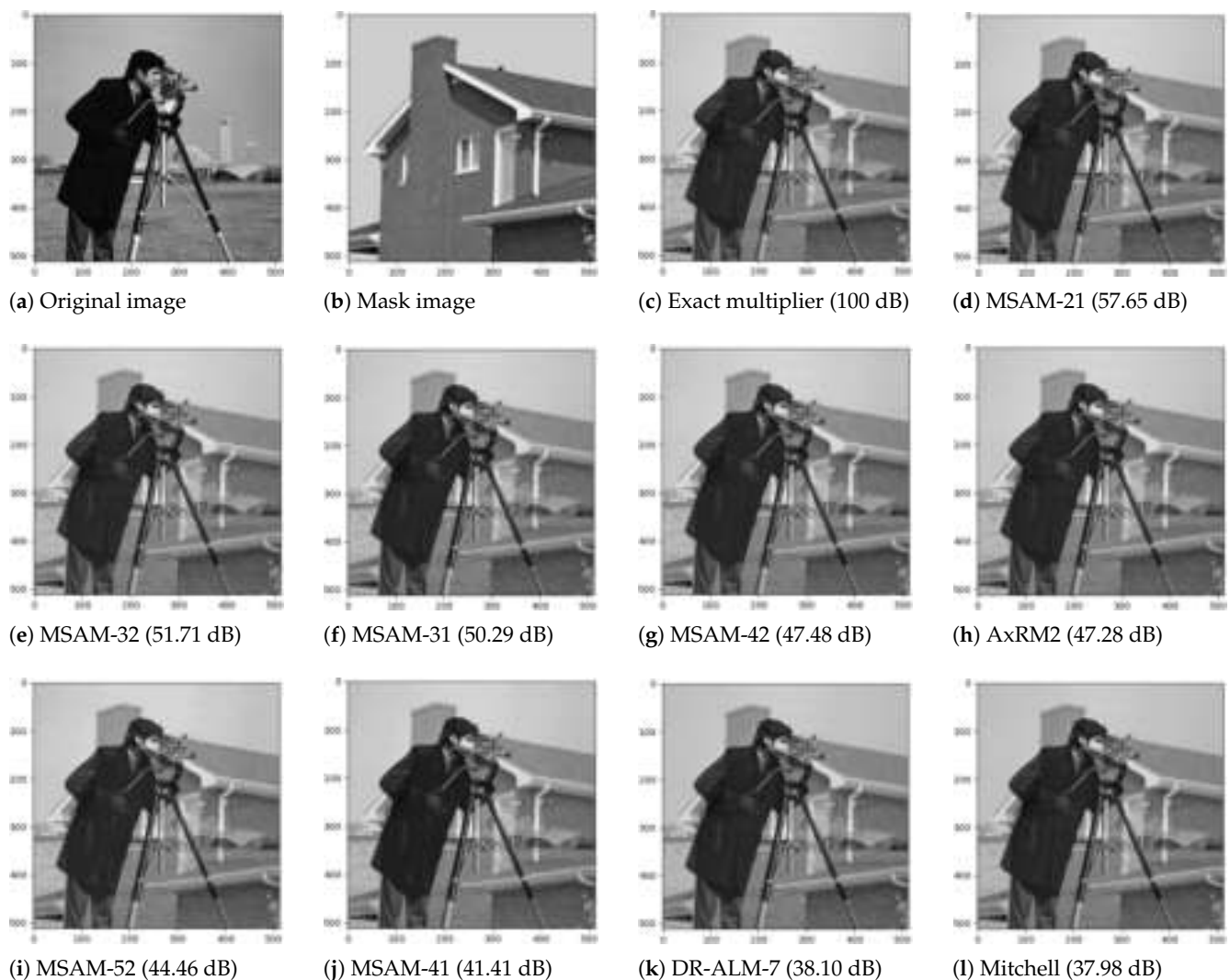
Table 5 presents the experimental results of the matrix multiplication module in the Transformer accelerator, highlighting the effective reduction in hardware overhead achieved with the proposed multiplier. The results demonstrate the multiplier's ability to optimize resource and energy usage when deploying LLMs on edge devices. Compared to the baseline result obtained using the exact multiplier, the MSAMs exhibit significant reductions in area, power consumption, and latency, with MSAM^Z-41 showing the most pronounced improvements. The employment of approximate computing has demonstrated the possibility of improving the efficiency of transformers by minimizing their computational complexity and memory demands [25]. Considering that large models have higher computational requirements, this suggests that more approximate modules can be included. Therefore, our future research should continue to explore and delve into a more appropriate and rational integration of these MSAMs, aiming to lower energy consumption while maintaining acceptable accuracy.

Table 5. Hardware performance of the proposed design with the transformer accelerator.

	Area (μm^2)	Comb. Power (mW)	Delay (ns)
Baseline	90,391.64	2.07	0.89
MSAM-41	67,618.91 (−25.19%)	1.82 (−12.08%)	0.79 (−11.24%)
MSAM ^Z -41	47,618.35 (−47.32%)	1.46 (−29.47%)	0.75 (−15.73%)
MSAM ^C -41	51,618.67 (−42.89%)	1.55 (−25.12%)	0.76 (−14.61%)

4.4. Case Study: Image Matrix Multiplication Accelerator

In our approach, we conduct a pixel-by-pixel matrix multiplication between an input image and a mask image, where the value of the corresponding pixel in the output image is determined according to the resulting product. Figure 11 shows the image multiplication results, where the image quality is positively correlated with the peak signal-to-noise ratio (PSNR); as such, the PSNR values for the generated images are included in their respective subtitles. The results obtained with most of our MSAMs indicate high quality, demonstrating a very good PSNR when compared to the exact multiplier in Figure 11c. Due to the relatively uniform distribution of the image data, as our design incorporates ESM, it has high utility and robustness in the context of image processing.

**Figure 11.** The results of multiplying (a) an input image with (b) a mask image using the selected multipliers.

4.5. Discussion on the Proposed MSAMs

4.5.1. Advantages and Limitations

The proposed MSAMs offer considerable energy savings while constraining the errors of approximate multipliers through the employment of most significant bit-guided approximation strategies. These strategies provide several primary advantages over other works. Firstly, MSAMs employ most significant bit-guided approximation techniques, facilitating multiplication to be carried out via shifting. This results in significant decreases in both area and energy consumption, making our approach highly efficient for neural network applications. Secondly, unlike logarithmic approaches, our approach guarantees that the approximate output is not one-sidedly biased. This feature enables compensation among different approximations, leading to enhanced overall accuracy and reliability. Furthermore, a key benefit of our approximation strategy is that it focuses on only one operand in the multiplication operation while fully preserving the precision of the other operand. This distinctive approach effectively ensures precision, providing a significant superiority over most other works in the field. The precise control over the approximation strategies and the ability to preserve precision in one operand contribute to the improved performance in neural network applications.

One major limitation of the proposed MSAMs is the relatively high error probability caused by the usage of pre-defined components to replace specific bits. This would render our multipliers as not applicable in scenarios where demanding precision is needed. Despite this limitation, the proposed MSAMs possess considerable practical worth, as the scale of the errors is not substantial, and given that the errors can be both positive and negative, the overall error is acceptable for batch operations.

4.5.2. Future Work

Matrix multiplication serves as the fundamental building element in the transformer architecture [25]. In the future, we aim to further explore the design of approximate multipliers with concentrated errors and establish an approximate multiplier library adapted for LLMs. Software-level approximations, like lower-complexity networks and pruning, facilitate the energy-efficient acceleration of DNNs [23]. While both software and hardware approximation techniques independently offer substantial energy benefits for a DNN, we intend to combine these techniques into a synergistic framework in future endeavors to maximize energy savings. We hope that this article will attract more researchers to conduct research in this aspect, allowing the application of MSAMs in more complex scenarios with tight energy budgets and strict accuracy requirements.

5. Conclusions

In this study, we proposed an energy-efficient most significant bit-guided dynamic approximate multiplier for neural network acceleration. Using a weight approximation strategy tailored for NNs and LLMs ultimately resulted in a high-efficiency approximate multiplier grounded in MSO shifting principles. The resultant MSAM design adeptly achieves a harmonious balance between precision and hardware cost. When applied to both LLMs and DNNs, our approach led to substantial reductions in hardware resource overheads. In comparison to the traditional exact multipliers used in LLMs and DNNs, our design achieved a remarkable 60% reduction in energy consumption without any significant adverse impacts on task accuracy. Furthermore, we conducted a case study on image merging, and the results demonstrate that our multipliers can be used to generate high-quality images. Compared with the exact multiplier, our approach performed very well when using the PSNR as an evaluation metric.

Author Contributions: Conceptualization, P.H.; Software, B.G.; Validation, B.G.; Writing—original draft, P.H.; Writing—review & editing, K.C.; Supervision, C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by grants from the National Natural Science Foundation of China (62101252, and 62134002).

Data Availability Statement: Upon reasonable request, the corresponding author can provide the data supporting the findings of this study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Raiaan, M.A.K.; Mukta, M.S.H.; Fatema, K.; Fahad, N.M.; Sakib, S.; Mim, M.M.J.; Ahmad, J.; Ali, M.E.; Azam, S. A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges. *IEEE Access* **2024**, *12*, 26839–26874. [\[CrossRef\]](#)
2. Rasch, M.J.; Mackin, C.; Gallo, M.L.; Chen, A.; Fasoli, A.; Odermatt, F.; Li, N.; Nandakumar, S.R.; Narayanan, P.; Tsai, H.; et al. Hardware-aware training for large-scale and diverse deep learning inference workloads using in-memory computing-based accelerators. *Nat. Commun.* **2023**, *14*, 5282. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Towhid, A.; Omid, R.; Mohammadi, K. On the Design of Iterative Approximate Floating-Point Multipliers. *IEEE Trans. Comput.* **2023**, *72*, 1623–1635. [\[CrossRef\]](#)
4. Zhang, H.; Ko, S.B. Efficient Approximate Posit Multipliers for Deep Learning Computation. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2023**, *13*, 201–211. [\[CrossRef\]](#)
5. Sayadi, L.; Timarchi, S.; Sheikh-Akbari, A. Two Efficient Approximate Unsigned Multipliers by Developing New Configuration for Approximate 4:2 Compressors. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2023**, *70*, 1649–1659. [\[CrossRef\]](#)
6. Schaefer, C.J.; Taheri, P.; Horeni, M.; Joshi, S. The Hardware Impact of Quantization and Pruning for Weights in Spiking Neural Networks. *IEEE Trans. Circuits Syst. II Express Briefs* **2023**, *70*, 1789–1793. [\[CrossRef\]](#)
7. Pinos, M.; Mrazek, V.; Vaverka, F.; Vasicek, Z.; Sekanina, L. Acceleration Techniques for Automated Design of Approximate Convolutional Neural Networks. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2023**, *13*, 212–224. [\[CrossRef\]](#)
8. Ahmadinejad, M.; Moaiyeri, M.H. Energy- and Quality-Efficient Approximate Multipliers for Neural Network and Image Processing Applications. *IEEE Trans. Emerg. Top. Comput.* **2022**, *10*, 1105–1116. [\[CrossRef\]](#)
9. Li, Z.; Zheng, S.; Zhang, J.; Lu, Y.; Gao, J.; Tao, J.; Wang, L. Adaptable Approximate Multiplier Design Based on Input Distribution and Polarity. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2022**, *30*, 1813–1826. [\[CrossRef\]](#)
10. Amirafshar, N.; Baroughi, A.S.; Shahhoseini, H.S.; TaheriNejad, N. Carry Disregard Approximate Multipliers. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2023**, *70*, 4840–4853. [\[CrossRef\]](#)
11. Lotrič, U.; Pilipović, R.; Bulić, P. A Hybrid Radix-4 and Approximate Logarithmic Multiplier for Energy Efficient Image Processing. *Electronics* **2021**, *10*, 1175. [\[CrossRef\]](#)
12. Faraone, J.; Kumm, M.; Hardieck, M.; Zipf, P.; Liu, X.; Boland, D.; Leong, P.H. AddNet: Deep neural networks using FPGA-optimized multipliers. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2019**, *28*, 115–128. [\[CrossRef\]](#)
13. Gong, B.; Chen, K.; Huang, P.; Wu, B.; Liu, W. Most Significant One-Driven Shifting Dynamic Efficient Multipliers for Large Language Models. In Proceedings of the 2024 IEEE International Symposium on Circuits and Systems (ISCAS), Singapore, 19–22 May 2024; pp. 1–5.
14. Mitchell, J.N. Computer Multiplication and Division Using Binary Logarithms. *IRE Trans. Electron. Comput.* **1962**, *EC-11*, 512–517.
15. Yin, P.; Wang, C.; Waris, H.; Liu, W.; Han, Y.; Lombardi, F. Design and analysis of energy-efficient dynamic range approximate logarithmic multipliers for machine learning. *IEEE Trans. Sustain. Comput.* **2020**, *6*, 612–625. [\[CrossRef\]](#)
16. Hashemi, S.; Bahar, R.I.; Reda, S. DRUM: A dynamic range unbiased multiplier for approximate applications. In Proceedings of the 2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, USA, 2–6 November 2015; pp. 418–425.
17. Waris, H.; Wang, C.; Xu, C.; Liu, W. AxRMs: Approximate recursive multipliers using high-performance building blocks. *IEEE Trans. Emerg. Top. Comput.* **2021**, *10*, 1229–1235. [\[CrossRef\]](#)
18. Strollo, A.G.M.; Napoli, E.; De Caro, D.; Petra, N.; Saggese, G.; Di Meo, G. Approximate multipliers using static segmentation: Error analysis and improvements. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2022**, *69*, 2449–2462. [\[CrossRef\]](#)
19. Park, G.; Kung, J.; Lee, Y. Simplified Compressor and Encoder Designs for Low-Cost Approximate Radix-4 Booth Multiplier. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *70*, 1154–1158. [\[CrossRef\]](#)
20. Jiang, H.; Santiago, F.J.H.; Mo, H.; Liu, L.; Han, J. Approximate Arithmetic Circuits: A Survey, Characterization, and Recent Applications. *Proc. IEEE* **2020**, *108*, 2108–2135. [\[CrossRef\]](#)
21. Mo, H.; Wu, Y.; Jiang, H.; Ma, Z.; Lombardi, F.; Han, J.; Liu, L. Learning the Error Features of Approximate Multipliers for Neural Network Applications. *IEEE Trans. Comput.* **2024**, *73*, 842–856. [\[CrossRef\]](#)
22. Danopoulos, D.; Zervakis, G.; Siozios, K.; Soudris, D.; Henkel, J. AdaPT: Fast Emulation of Approximate DNN Accelerators in PyTorch. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2023**, *42*, 2074–2078. [\[CrossRef\]](#)
23. Sanyal, S.; Negi, S.; Raghunathan, A.; Roy, K. Approximate Computing for Machine Learning Workloads: A Circuits and Systems Perspective. In *Approximate Computing*; Liu, W., Lombardi, F., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 365–395.

24. Dettmers, T.; Lewis, M.; Belkada, Y.; Zettlemoyer, L. GPT3.int8(): 8-bit Matrix Multiplication for Transformers at Scale. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022.
25. Danopoulos, D.; Zervakis, G.; Soudris, D.; Henkel, J. TransAxx: Efficient Transformers with Approximate Computing. *arXiv* **2024**, arXiv:2402.07545.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

A Single-Stage Electronic Lighting Driver Circuit Utilizing SiC Schottky Diodes for Supplying a Deep Ultraviolet LED Disinfection and Sterilization Lamp

Chun-An Cheng , Chien-Hsuan Chang , Hung-Liang Cheng , En-Chih Chang *, Long-Fu Lan, Sheng-Hong Hou and Cheng-Kuan Lin

Department of Electrical Engineering, I-Shou University, Kaohsiung City 84001, Taiwan; cacheng@isu.edu.tw (C.-A.C.); chchang@isu.edu.tw (C.-H.C.); hlcheng@isu.edu.tw (H.-L.C.); isu11001004m@cloud.isu.edu.tw (L.-F.L.); isu11001001m@cloud.isu.edu.tw (S.-H.H.); isu11001014m@cloud.isu.edu.tw (C.-K.L.)

* Correspondence: enchihchang@isu.edu.tw; Tel.: +886-7-6577711 (ext. 6642)

Abstract: Recently, a new type of lighting source, deep ultraviolet light-emitting diode (LED), has appeared in the markets of space purification and surface sterilization. In this paper, a new type of electronic lighting driver for supplying a deep-ultraviolet LED sterilization lamp is proposed and developed. The main circuit combines a buck converter and a flyback converter into a single-stage single-switch buck-flyback AC-DC power converter with power factor correction. In addition, the proposed electronic lighting driver leverages a wide bandgap SiC Schottky diode as the output diode to lower the power diode losses and recycles the energy stored in the leakage inductance of the transformer in order to improve the circuit efficiency. The magnetizing inductor inside the presented AC-DC power converter is designed to operate in discontinuous conduction mode (DCM), which naturally enables power factor correction (PFC). A single-stage prototype driver with a power rating of 3.6 W (90 V/40 mA) was developed and implemented for providing a deep ultraviolet LED disinfection and sterilization lamp. Experimental results show that the measured power factor (PF) is greater than 0.9 and the measured total harmonic distortion (THD) of the input current is less than 18% at an input utility voltage of 110 V. Furthermore, the measured output voltage ripple factor is less than 1% and the output current ripple factor is less than 4%. In addition, the proposed single-stage electronic lighting driver for supplying a deep ultraviolet LED disinfection and sterilization lamp achieves high circuit efficiency (greater than 90%), low circuit component count, and low circuit cost.

Keywords: electronic lighting driver circuit; deep ultraviolet LED; disinfection and sterilization lamp; power factor correction; total harmonic distortion



Citation: Cheng, C.-A.; Chang, C.-H.; Cheng, H.-L.; Chang, E.-C.; Lan, L.-F.; Hou, S.-H.; Lin, C.-K. A Single-Stage Electronic Lighting Driver Circuit Utilizing SiC Schottky Diodes for Supplying a Deep Ultraviolet LED Disinfection and Sterilization Lamp. *Electronics* **2024**, *13*, 3048. <https://doi.org/10.3390/electronics13153048>

Academic Editors: Yimeng Zhang, Lejia Sun and Yuming Zhang

Received: 24 June 2024

Revised: 27 July 2024

Accepted: 29 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Microorganisms such as bacteria, viruses, cysts and molds use germination or spore production to reproduce. When microorganisms are irradiated or exposed to disinfecting and sterilizing lamps, the light penetrates through the cell walls and destroys their DNA, changing their DNA so they become harmless and incapable of reproducing, or even die, thus achieving the effect of disinfecting and sterilizing. Sterilization and disinfection lamps are used in a wide range of applications in a variety of sectors, including healthcare facilities, the food and beverage industry, public spaces and transportation, commercial and office buildings, residential buildings, hospitality, educational institutions, retail and grocery stores, water treatment and agriculture. These diverse applications highlight the versatility and effectiveness of disinfecting and sterilizing lamps in promoting health and safety in different environments [1–4]. The traditional light source for sterilization and disinfection is the ultraviolet (UV) mercury lamp, which has the advantages of low price and high luminous power and is commonly used in large places such as hospitals, factories,

restaurants and schools. However, the disadvantages of UV mercury lamps are long warm-up time, short lamp life, and that the mercury content in the lamps does not meet environmental requirements. UV light is invisible to the human eye and has a wavelength of 100 to 400 nanometers. Depending on the wavelength, UV can be subdivided into three types: long-wave UV-A (wavelengths of 320–400 nm), medium-wave UV-B (wavelengths of 280–320 nm) and short-wave UV-C (wavelengths of 100–280 nm). Short-wave UV-C, which can also be referred to as deep ultraviolet, has disinfecting and germicidal properties. The use of short-wave ultraviolet UV-C or deep ultraviolet light destroys the DNA or RNA molecular structure of microbial cells, thereby preventing cell growth or regeneration, and ultimately achieving the effect of disinfection and sterilization. Deep UV LEDs are characterized by environmental protection and energy saving, long life, no warm-up time, no mercury, and miniaturization of light source. Traditional UV mercury lamps have been gradually replaced by deep UV LEDs, which have become a new light source for space purification and surface sterilization markets such as homes, schools, hospitals, office buildings, supermarkets or department stores. In addition, deep ultraviolet disinfection and sterilization lamps can easily kill bacteria, but due to the low penetrating power of deep ultraviolet disinfection and sterilization lamps, they can be affected by dust particles; if the cells damaged by deep ultraviolet disinfection and sterilization lamps are not completely destroyed, the cells are still able to be repaired and revitalized. Therefore, it is important to provide the appropriate distance and sufficient irradiation time and intensity of the deep ultraviolet disinfection and sterilization lamp to the disinfected objects in order to achieve a good sterilization effect [5–8].

Figure 1 shows the two-stage circuit topology of an existing electronic lighting driver for supplying deep ultraviolet LED sterilization and disinfection lamps. The first stage is an AC-DC power converter with power factor correction (PFC), and the second stage is a DC-DC converter and delivers the required DC power to the deep-UV LED sterilization lamp [8]. Two-stage electronic lighting driver circuits that can be applied to deep ultraviolet light-emitting diode (LED) sterilization lamps require separate controllers for both the front and rear stages, making the circuits more complex and requiring a larger number of circuit components. In order to reduce the number of power switches and circuit components and to improve the overall efficiency of the conventional two-stage version of the circuit, the literature [9–20] has developed a number of single-stage, single-switch LED driver circuits that integrate an AC-DC power converter and a DC-DC power converter suitable for powering deep ultraviolet LED sterilization and disinfection lamps.

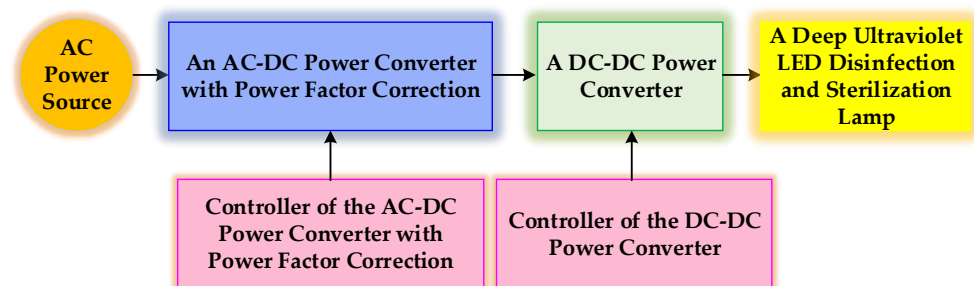


Figure 1. Block diagram of the existing two-stage electronic lighting driver for supplying a deep ultraviolet LED disinfection and sterilization lamp [8].

As the environmental issues derived from global warming and carbon emissions have become more and more serious in recent years, human beings have taken energy saving, carbon reduction, and love for the earth as the common primary development direction. Thus, fossil fuels must be gradually reduced and rapidly introduced into green energy and power saving applications, and, therefore, high energy efficiency and low energy consumption are also gradually adopted as the goal in daily life products. In the Paris Agreement of the United Nations Conference on Climate Change, it is declared that the

global warming rate must be kept within 2 °C. Based on the current economic development trend, even if the warming is kept within 2 °C in 2050, CO₂ emissions will still increase by 21%, and up to 50% more electricity will have to be generated to cope with and provide for various human activities. Therefore, it is increasingly important to significantly upgrade and improve existing energy sources. The largest raw material for semiconductors is the production of first-generation silicon (Si) wafers. However, the physical properties of this material have reached their limits in existing Si-based products and it is no longer possible to increase power, reduce heat loss and increase speed. As a result, there is a need to evolve to other materials that can better utilize the efficiency of electron transfer and low energy consumption, and it is in this context that the third-generation wide-bandgap semiconductors that are currently hotly debated in the marketplace, namely, silicon carbide (SiC) and gallium nitride (GaN), which are characterized by high energy efficiency and low energy consumption, have come into being. In addition, power devices made of wide-gap semiconductor materials can operate at higher voltages, generate more power, and have lower energy losses, while being significantly smaller than power devices made of silicon semiconductor materials. Silicon carbide (SiC) is a compound semiconductor material consisting of silicon (Si) and carbon (C). SiC has a lower resistance of the drift layer than Si, does not require conduction modulation, has an energy gap about 3-times wider than that of Si, and has an insulation breaking field strength about 10-times higher than that of Si. Compared to the high-voltage Si-PN diodes and high-speed Si-Fast Recovery diodes, SiC Schottky diodes made of wide-gap semiconductor materials have excellent high-speed and high-voltage characteristics. Since the reverse recovery time of SiC Schottky diodes is extremely short, the energy lost during reverse recovery can be greatly reduced, which is suitable for applications with high switching frequency, and is conducive to the miniaturization of the overall circuit as well as the improvement of power density [21,22]. Supplementary notes on the long-term reliability and potential degradation of silicon carbide Schottky diodes under thermal and electrical stresses are given below. According to the literature [21], silicon carbide Schottky diodes do not have minority carrier injection and their recovery time does not increase with temperature compared to conventional silicon diodes. In addition, SiC Schottky diodes are majority carrier devices and therefore the reverse current is smaller and temperature independent. In addition, compared to silicon diodes, SiC Schottky diodes have the advantage of faster recovery, which reduces switching losses and noise, and lower forward voltage, which reduces conduction losses. Furthermore, the literature [22] mentions the long-term reliability of SiC Schottky diodes with respect to dV/dt failure and dI/dt failure. Furthermore, in silicon diodes, if dI/dt is large, a pattern of increased recovery current (I_{rr}) occurs, which can damage the device through current concentration. In contrast, since the recovery current in SiC Schottky diodes is very small, it can be assumed that such a pattern is unlikely to occur.

To address the above issues, a novel electronic lighting driver for providing a deep ultraviolet LED disinfection and sterilization lamp is proposed and developed in this paper, which is an extension and further improvement of the literature [20]. The main circuit combines a buck converter and a flyback converter into a single-stage single-switch buck-flyback AC-DC power converter with input-current shaping. In addition, the proposed driver recycles the energy stored in the leakage inductance of the transformer and exploits a wide bandgap silicon carbide Schottky diode as the output diode to enhance the circuit efficiency.

This paper is organized and introduced as follows. Section 2 describes and analyzes the mode of operation of a single-stage electronic lighting driver circuit utilizing a SiC Schottky diode to supply a deep ultraviolet LED disinfection and sterilization lamp, and provides design guidelines for some circuit parameters. Section 3 shows the experimental results of a prototype electronic lighting driver circuit utilizing a SiC Schottky diode to supply a deep ultraviolet LED sterilization and disinfection lamp. Finally, some conclusions are given in Section 4.

2. Descriptions and Analysis of Operational Modes in the Proposed Single-Stage Electronic Lighting Driver Circuit Utilizing SiC Schottky Diodes for Supplying a Deep Ultraviolet LED Disinfection and Sterilization Lamp

The electronic lighting driver with PFC for powering a deep ultraviolet LED disinfection and sterilization lamp developed in this paper is shown in Figure 2, which integrates a buck converter and a flyback converter into a single-stage AC-DC power conversion and includes a filter inductor L_F , a filter capacitor C_F , a full-bridge rectifier (including D_1 , D_2 , D_3 and D_4), a power switch S_B , two diodes D_B and D_F , a transformer T_R with a magnetizing inductor L_M and a leakage inductor L_{lk} , two output capacitors C_{O1} and C_{O2} , and the deep ultraviolet LED disinfection and sterilization lamp. By designing the magnetizing inductor L_M in the proposed electronic lighting driver circuit to operate in discontinuous conduction mode (DCM), PFC can be realized naturally. In addition, the stored energy in the leakage inductance L_{lk} of the transformer T_R in the proposed electronic lighting driver circuit can be recovered and the circuit efficiency can be enhanced by exploiting a wide-bandgap SiC Schottky diode as an output diode.

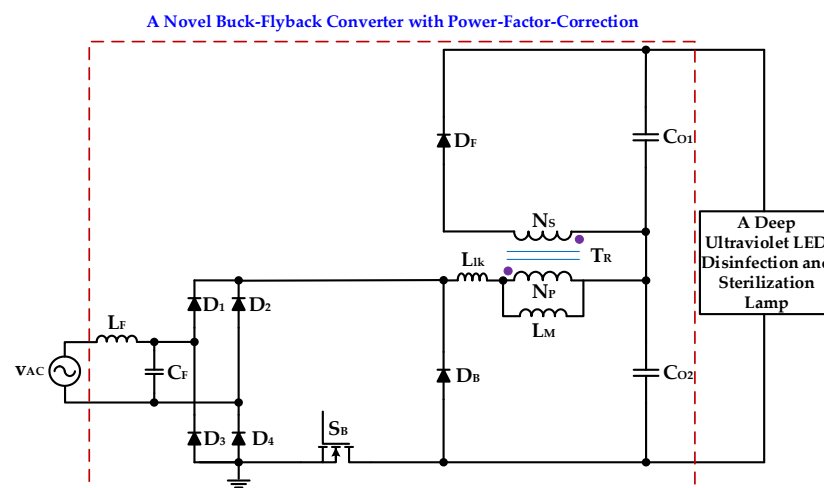


Figure 2. The proposed single-stage electronic lighting driver utilizing SiC Schottky diodes for supplying a deep ultraviolet LED disinfection and sterilization lamp.

Figure 3 is the equivalent diagram of the single-stage electronic lighting driver circuit utilizing SiC Schottky diodes for supplying a deep ultraviolet LED disinfection and sterilization lamp developed in this paper with the power factor correction function. Theoretical waveforms of the proposed single-stage electronic lighting driver utilizing SiC Schottky diodes for supplying a deep ultraviolet LED disinfection and sterilization lamp are shown in Figure 4. When analyzing the proposed single-stage electronic lighting driver circuit for the deep ultraviolet LED disinfection and sterilization lamp, assumptions are made during descriptions and explanations of the operation mode in the presented driver circuit and are shown as follows:

- The equivalent voltage source of the input utility-line voltage after passing through the full-wave rectifier circuit is denoted by V_{REC} . Since the switching frequency f_s is much larger than the line frequency f_{AC} , the rectified voltage V_{REC} can be regarded as a constant value during one switching cycle in the circuit mode analysis.
- The leakage inductance L_{lk} of the transformer T_R is considered.
- The magnetizing inductor L_M of the transformer T_R is designed to operate in discontinuous conduction mode (DCM).
- Neglecting the conduction voltage drops and their equivalent resistances of all diodes.
- The remaining circuit components are considered as ideal components.

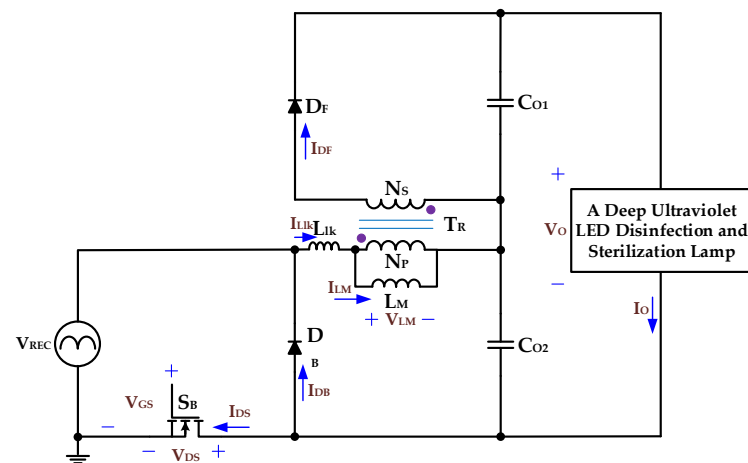


Figure 3. Equivalent circuit of the proposed electronic lighting driver circuit utilizing SiC Schottky diodes for supplying a deep ultraviolet LED disinfection and sterilization lamp while analyzing the operational modes.

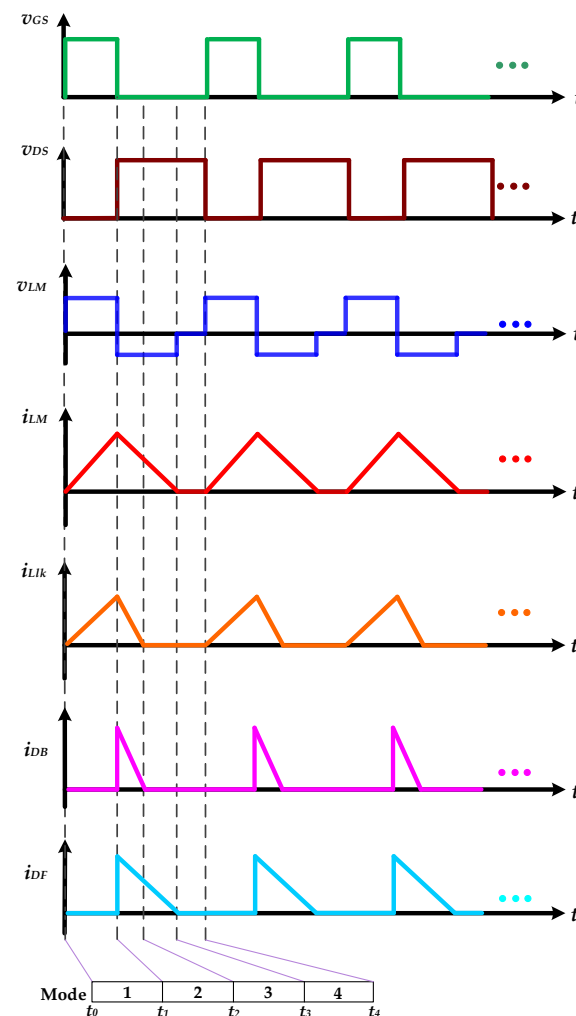


Figure 4. Theoretical waveforms of the proposed single-stage electronic lighting driver utilizing SiC Schottky diodes for supplying a deep ultraviolet LED disinfection and sterilization lamp.

Operation Mode 1 ($t_0 \leq t < t_1$): Figure 5 shows the equivalent circuit of the proposed single-stage electronic lighting driver circuit for supplying a deep ultraviolet LED disinfection and sterilization lamp during Mode 1. The power switch S_B is driven on, and the

equivalent voltage source V_{REC} is connected to both ends of the diode D_B , making the diode D_B a reverse biased non-conducting state. The equivalent voltage source V_{REC} provides energy to the output capacitor C_{O2} and the magnetizing inductor L_M of the transformer T_R as well as the leakage inductor through the switch S_B , which results in a linear increase in the current of the magnetizing inductor L_M , and the voltage polarity on both sides of the magnetization inductor L_M and the leakage inductor L_{lk} is positive left and negative right. The output capacitors C_{O1} and C_{O2} provide energy to the deep ultraviolet LED disinfection and sterilization lamp. When the switch S_B turns off, the current of the magnetizing inductor L_M reaches its maximum value and the mode ends.

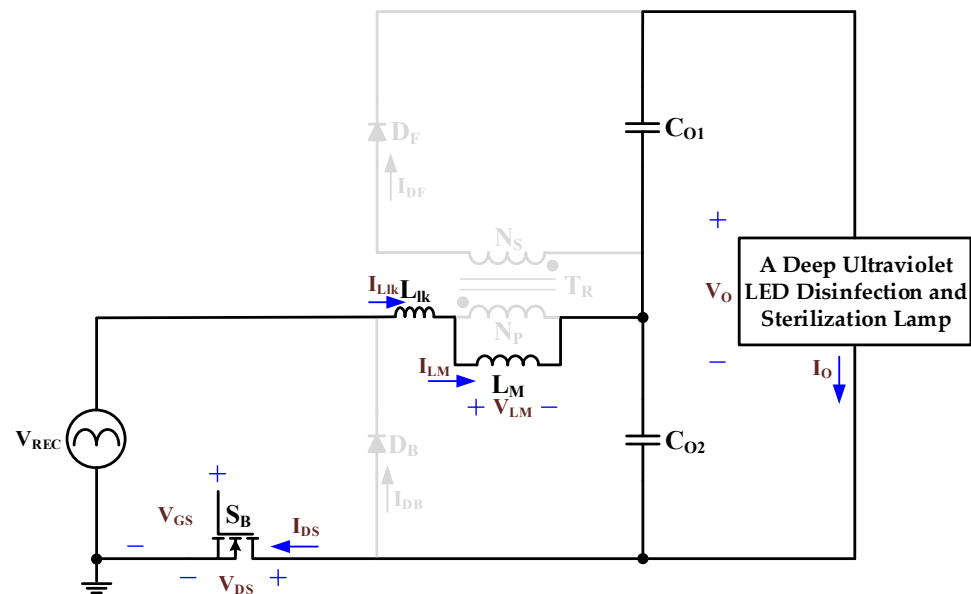


Figure 5. Equivalent circuit of the proposed single-stage electronic lighting driver circuit utilizing SiC Schottky diodes for supplying a deep ultraviolet LED disinfection and sterilization lamp during Mode 1.

Operational Mode 2 ($t_1 \leq t < t_2$): Figure 6 shows the equivalent circuit of the proposed single-stage electronic lighting driver circuit for supplying a deep ultraviolet LED disinfection and sterilization lamp during Mode 2. When the power switch S_B turns off, the voltage polarity of the magnetizing inductor L_M and the leakage inductor L_{lk} is changed to right positive and left negative according to the Lenz's law, which turns the diode D_B to be the on state of forward bias. The current of the magnetizing inductor L_M flows into the position where the primary-side winding N_P of the transformer T_R does not have a dot end, and the voltage polarity of the two ends of the winding N_P is right-positive and left-negative. Therefore, the polarity of the voltage reflected to both ends of the secondary-side winding N_S is positive left and negative right, which makes the diode D_F work in the forward-biased conduction state. At this time, the magnetizing inductor L_M and the leakage inductor L_{lk} are in the state of releasing energy, causing the current of the magnetizing inductor L_M to decrease linearly. Since the diode D_B is in the forward-biased conduction state, the energy of the magnetizing inductor L_M and the leakage inductor L_{lk} is supplied to the output capacitor C_{O2} through the diode D_B , and the current of the magnetizing inductor L_M shows a linear decrease. Due to the forward-biased conduction of the diode D_F , the energy of the magnetizing inductor L_M is supplied through the transformer T_R and the diode D_F to the output capacitance C_{O1} and C_{O2} as well as to the deep ultraviolet LED disinfection and sterilization lamp. In addition, the output capacitors C_{O1} and C_{O2} continue to provide energy to the deep ultraviolet LED disinfection and sterilization lamp. The mode ends when the leakage inductor L_{lk} finishes releasing energy and the current I_{Llk} of the leakage inductor L_{lk} drops linearly to zero.

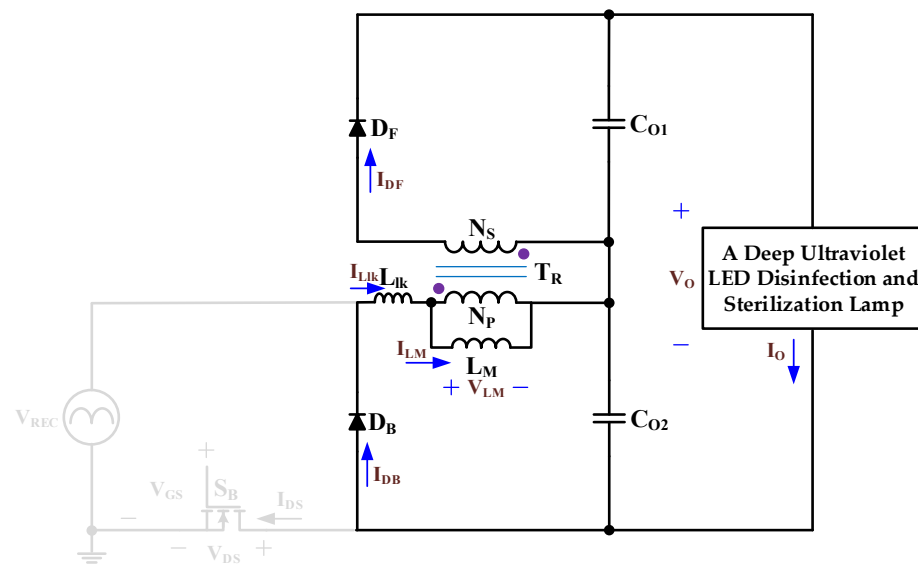


Figure 6. Equivalent circuit of the proposed single-stage electronic lighting driver circuit utilizing SiC Schottky diodes for supplying a deep ultraviolet LED disinfection and sterilization lamp during Mode 2.

Operational Mode 3 ($t_2 \leq t < t_3$): Figure 7 shows the equivalent circuit of the proposed single-stage electronic lighting driver circuit for supplying a deep ultraviolet LED disinfection and sterilization lamp during Mode 3. The energy of the magnetizing inductor L_M is continuously supplied through the transformer T_R and the diode D_F to the output capacitors C_{O1} and C_{O2} and to the deep ultraviolet LED disinfection and sterilization lamp. In addition, the output capacitors C_{O1} and C_{O2} continue to provide energy to the deep ultraviolet LED disinfection and sterilization lamp. This mode ends when the magnetizing inductor L_M has finished releasing energy and the magnetizing inductor current I_{LM} drops linearly to zero.

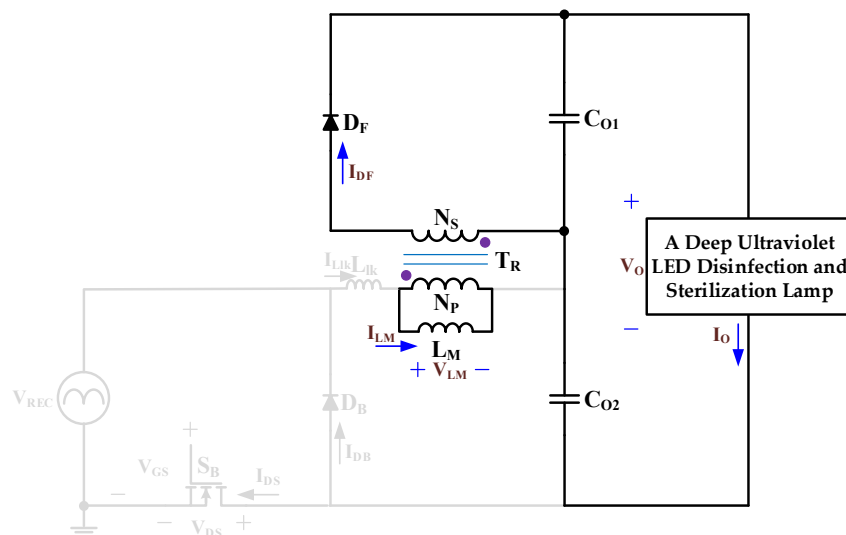


Figure 7. Equivalent circuit of the proposed single-stage electronic lighting driver circuit utilizing SiC Schottky diodes for supplying a deep ultraviolet LED disinfection and sterilization lamp during Mode 3.

Operational Mode 4 ($t_3 \leq t < t_4$): Figure 8 shows the equivalent circuit of the proposed single-stage electronic lighting driver circuit for supplying a deep ultraviolet LED disinfection and sterilization lamp during Mode 4. The output capacitors C_{O1} and C_{O2}

continuously provide energy to the deep ultraviolet LED disinfection and sterilization lamp. When the power switch S_B is turned on again, this mode ends and the circuit reverts to the first mode of operation.

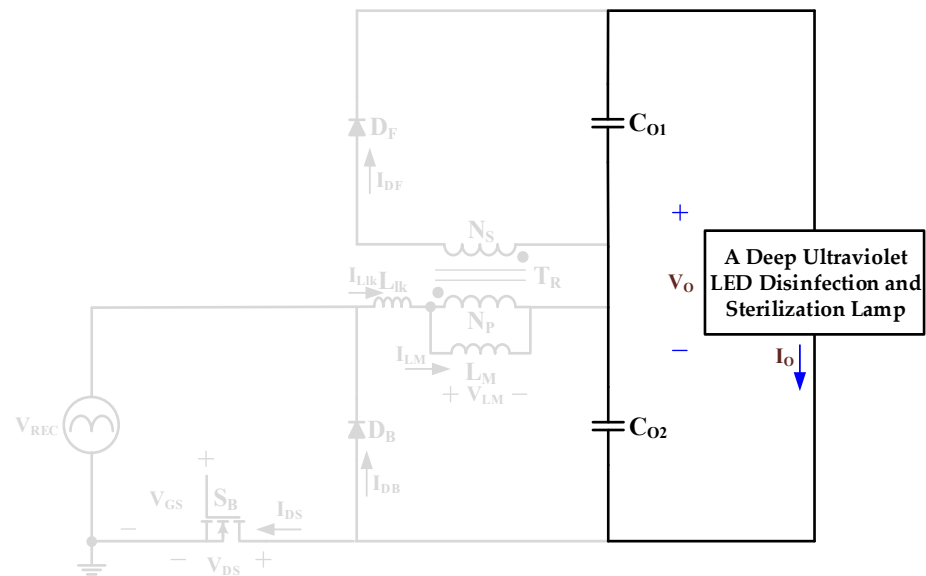


Figure 8. Equivalent circuit of the proposed single-stage electronic lighting driver circuit utilizing SiC Schottky diodes for supplying a deep ultraviolet LED disinfection and sterilization lamp during Mode 4.

2.1. Design Guideline of the Magnetizing Inductor L_M in the Transformer T_R

In order to show the design guideline of the magnetizing inductor L_M in the transformer T_R , Figure 9 illustrates theoretical waveforms of the magnetizing inductor current $i_{LM}(t)$, the peak level of $i_{LM-pk}(t)$ and the input utility-line current $i_{AC}(t)$ in the positive half-cycle of the utility-line voltage $v_{AC}(t)$.

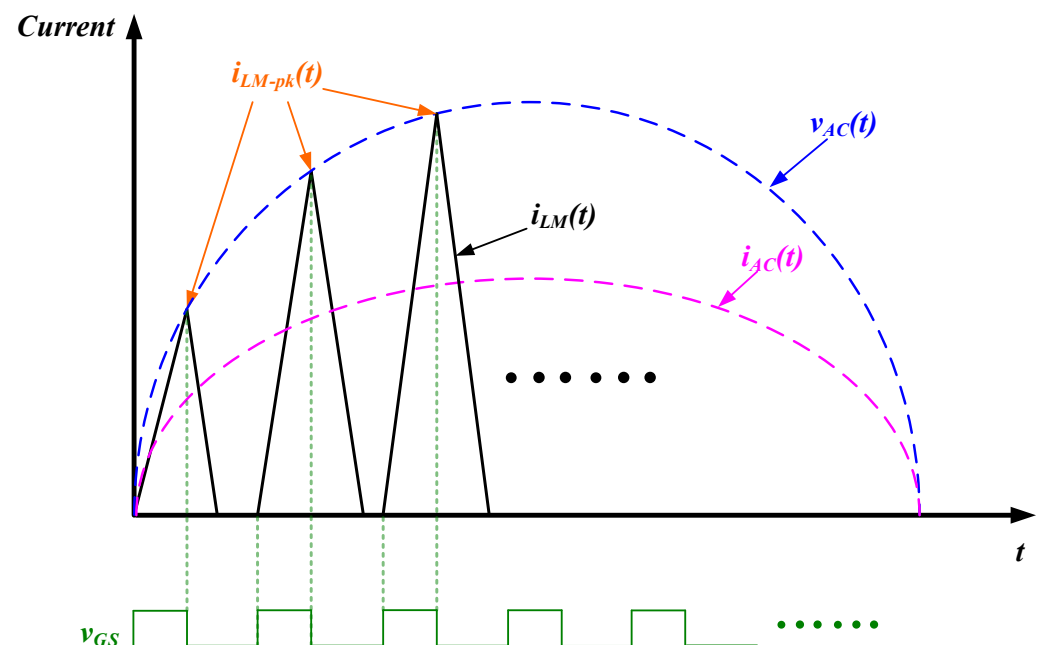


Figure 9. Illustrative waveforms of the magnetizing inductor current $i_{LM}(t)$, the peak level of $i_{LM-pk}(t)$ and the input utility-line current $i_{AC}(t)$ in the positive half-cycle of the utility-line voltage $v_{AC}(t)$.

Referring to Figure 9, the positive half-cycle of the instantaneous utility-line voltage $v_{AC}(t)$ can be expressed by

$$|v_{AC}(t)| = \sqrt{2}v_{AC-rms}|\sin(2\pi f_{AC}t)| \quad (1)$$

The peak level of the magnetizing inductor current i_{LM} , denoted as $i_{LM,peak}$, can be represented by

$$i_{LM,peak}(t) = \frac{\sqrt{2}v_{AC-rms}|\sin(2\pi f_{AC}t)|DutyT_S}{2L_M} \quad (2)$$

By filtering the high-frequency components of the peak level of the magnetizing inductor current $i_{LM,peak}(t)$, the input utility-line current i_{AC} is equal to the average level of $i_{LM,peak}(t)$ during one switching period and can be expressed as

$$i_{AC}(t) = \frac{1}{T_{AC}} \int_0^{T_{AC}} i_{LM,peak}(t)dt = \frac{\sqrt{2}v_{AC-rms}Duty^2T_S}{4L_M} \sin(2\pi f_{AC}t) \quad (3)$$

where T_{AC} is the utility-line period.

The average value of the input utility line power P_{IN} is obtained by multiplying the instantaneous value of the utility line voltage $v_{AC}(t)$ by the instantaneous value of the utility line current $i_{AC}(t)$, and then averaging it over a period of one cycle, which is calculated as follows

$$P_{IN} = \frac{1}{T_{AC}} \int_0^{T_{AC}} v_{AC}(t)i_{AC}(t)dt = \frac{v_{AC-rms}^2Duty^2T_S}{4L_M} \quad (4)$$

The rated output power P_O of the deep ultraviolet LED disinfection and sterilization lamp is related to the estimated efficiency of the driver circuit η multiplied by the input power P_{IN} , which is expressed as follows

$$P_O = \eta P_{IN} = \frac{\eta v_{AC-rms}^2Duty^2T_S}{4L_M} \quad (5)$$

Rearranging (5), the design formula for the inductance of the magnetizing inductor L_M is given by

$$L_M = \frac{\eta v_{AC-rms}^2Duty^2T_S}{4P_O} \quad (6)$$

2.2. Design Guideline of the Output Capacitors C_{O1} and C_{O2}

The design consideration of the output capacitors, C_{O1} and C_{O2} , depends on the DC output voltage, the allowable overvoltage, the output power and the desired voltage ripple. The voltage ripple, ΔV_O , is half the peak-to-peak value of the output voltage at twice the mains frequency, and is a function of the capacitance C_O and the peak capacitor current (which is equal to the output current I_O) and can be expressed in the following equation [23].

$$\Delta V_O = I_O \sqrt{\left(\frac{1}{2\pi \times 2f_{AC} \times C_O}\right)^2 + R_{ESR}^2} \quad (7)$$

Neglecting the equivalent series resistance, which is denoted as R_{ESR} , of the output capacitor, the capacitance C_O of the output capacitor can be expressed as

$$C_O \geq \frac{I_O}{4\pi f_{AC} \times \Delta V_O} = \frac{P_O}{4\pi f_{AC} \times V_O \times \Delta V_O} \quad (8)$$

In addition, the voltage ripple ΔV_O is typically selected from 1% to 5% of the output voltage.

3. Experimental Results of the Proposed Electronic Lighting Driver Circuit Utilizing SiC Schottky Diodes for Supplying a Deep Ultraviolet LED Disinfection and Sterilization Lamp

Figure 10 shows a photograph of the deep ultraviolet LED disinfection and sterilization module used in this paper with specifications that include a wavelength of 275 nm and a point angle of 120 degrees. Moreover, the manufacturer has connected a constant current control IC in series and a Schottky diode in parallel for reverse polarity protection on the aluminum substrate in each deep ultraviolet LED disinfection and sterilization module. The specification of the deep ultraviolet LED disinfection and sterilization lamp used in this paper consists of ten deep ultraviolet LED disinfection and sterilization modules connected in series with a rated voltage of 90 volts, a rated current of 40 milliamps, and a rated power of 3.6 watts. Table 1 shows a comparison between the deep ultraviolet LED disinfection and sterilization module in [24] and the module used in this paper. As can be seen from Table 1, the deep ultraviolet LED disinfection and sterilization module used in this paper has a larger peak wavelength, a slightly narrower viewing angle, and a slightly larger optical output power and spectral half-width than the module in [24]. Table 2 shows the electrical parameters of the single-stage electronic lighting prototype driver circuit for the deep ultraviolet LED disinfection and sterilization lamp developed in this paper. The RMS AC input voltage was 110 volts, and the specifications of the deep ultra LED disinfection and sterilization lamp were used as the output parameters, including: output power of 3.6 watts, output voltage of 90 volts, and output current of 40 milliamps. Moreover, safety concerns and ensuring compliance with regulatory standards for UV exposure and electrical safety include: Do not look directly at the UV light source when using a deep ultraviolet disinfection and sterilization lamp as deep UV radiation may cause skin erythema, conjunctival irritation and fatigue. In addition, do not overuse deep UV disinfection and sterilization lamps as prolonged use or contact with UV lamps or direct exposure to UV radiation may cause skin and eye damage.

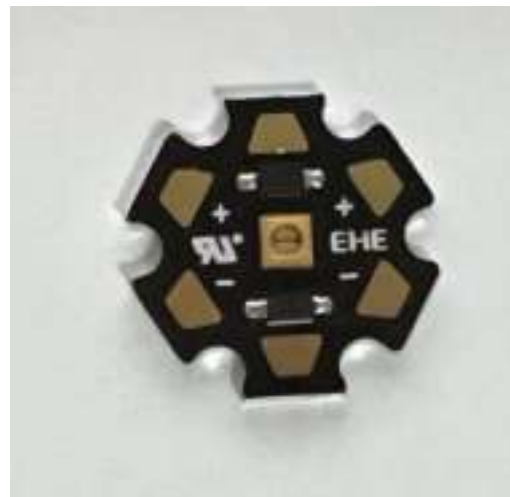


Figure 10. Photograph of the deep ultraviolet LED disinfection and sterilization module used in this paper with specifications that include a wavelength of 275 nm and a point angle of 120 degrees.

3.1. Calculating the Magnetizing Inductor L_M in the Transformer T_R

Referring to (6) with a η of 0.8, a *Duty* of 0.25, a P_O of 3.6 W, a switching frequency f_S of 50 kHz (which is the reciprocal of the switching period T_S), and a v_{AC-rms} of 110 V, the inductance of the magnetizing inductor L_M is calculated by

$$L_M = \frac{\eta v_{AC-rms}^2 \text{Duty}^2 T_S}{4P_O} = \frac{0.8 \times 110^2 \times 0.25^2 \times \left(\frac{1}{50,000}\right)}{4 \times 3.6} = 834 \mu H \quad (9)$$

Table 1. Comparisons between the deep ultraviolet LED disinfection and sterilization module in [24] and the one utilized in this paper.

Item	The Deep Ultraviolet LED Disinfection and Sterilization Module in [24]	The Deep Ultraviolet LED Disinfection and Sterilization Module Utilized in this Paper
Typical value of the peak wavelength	255 nm	275 nm
Typical value of the optical output power	3.5 mW	4 mW
viewing angle	125 degree	120 degree
Spectrum half width	11 nm	12 nm

Table 2. Specifications of the proposed electronic lighting driver circuit utilizing SiC Schottky diodes for supplying a deep ultraviolet LED disinfection and sterilization lamp.

Parameter	Value
Input AC voltage v_{AC}	110 V
Rated output power P_O	3.6 W
Rated output voltage V_O	90 V
Rated output current I_O	40 mA

In order to operate the magnetizing inductor current in a discontinuous conduction mode so as to allow the driver circuit to have an input current shaping function and to improve the power factor, the magnetizing inductor L_M is selected as 800 μ H for the prototype driver circuit.

3.2. Calculating of the Output Capacitors C_{O1} and C_{O2}

Substituting the circuit parameters into Equation (8), with a P_O of 3.6 W, an f_{AC} of 60 Hz, a V_O of 90 V, a ΔV_O of 2.7 V (which is 3% of the output voltage V_O), the capacitance C_O can be obtained by:

$$C_O \geq \frac{P_O}{4\pi f_{AC} \times V_O \times \Delta V_O} = \frac{3.6}{4\pi \times 60 \times 90 \times 2.7} = 196.49 \mu F \quad (10)$$

In implementing the prototype circuit, the capacitance values of the output capacitors C_{O1} and C_{O2} were selected to be 220 μ F and the withstand voltage to be 250 V.

Table 3 shows the component specifications of the single-stage electronic lighting prototype driver circuit for the deep ultraviolet LED disinfection and sterilization lamp developed in this paper. In addition, majority carrier diodes with Schottky technology on silicon carbide wide bandgap material have higher performance and are used as output diodes D_B and D_F . By varying the number of series-connected deep ultraviolet disinfection and sterilization modules used, a scalable design of the proposed electronic lighting driver circuits can be achieved to accommodate higher power applications and varying load conditions. As a result, the output power P_O and output voltage V_O can be varied while the output current I_O remains constant and can be used in a variety of practical application scenarios. Key circuit parameters of the proposed drive circuit, such as the magnetizing inductor and output capacitor, can then be redesigned.

Figure 11 shows the measured waveform of the magnetizing inductor current i_{LM} . Figure 12 shows the measured unfolded waveform of the magnetizing inductor current i_{LM} . From Figure 12, it can be seen that the magnetizing inductor current i_{LM} operates in the discontinuous conduction mode. Figure 13 shows the measured waveforms of output voltage V_O and output current I_O , and the average values of output voltage and current were 91.35 volts and 38.57 milliamps, respectively. Figure 14 shows the ripple waveforms of output voltage V_O and output current I_O .

Table 3. Key components used in the proposed electronic lighting driver circuit utilizing SiC Schottky diodes for supplying a deep ultraviolet LED disinfection and sterilization lamp.

Component	Value
Filter inductor L_f	1 mH
Filter capacitor C_f	220 nF
Diodes D_1, D_2, D_3, D_4	MUR460
Power switch S_B	IRF730
Magnetizing inductor L_M	800 μ H
Leakage inductor L_{lk}	12 μ H
Diodes D_B, D_F	VS-3C04ET07T-M3
Output capacitor C_{O1}, C_{O2}	220 μ F/200 V

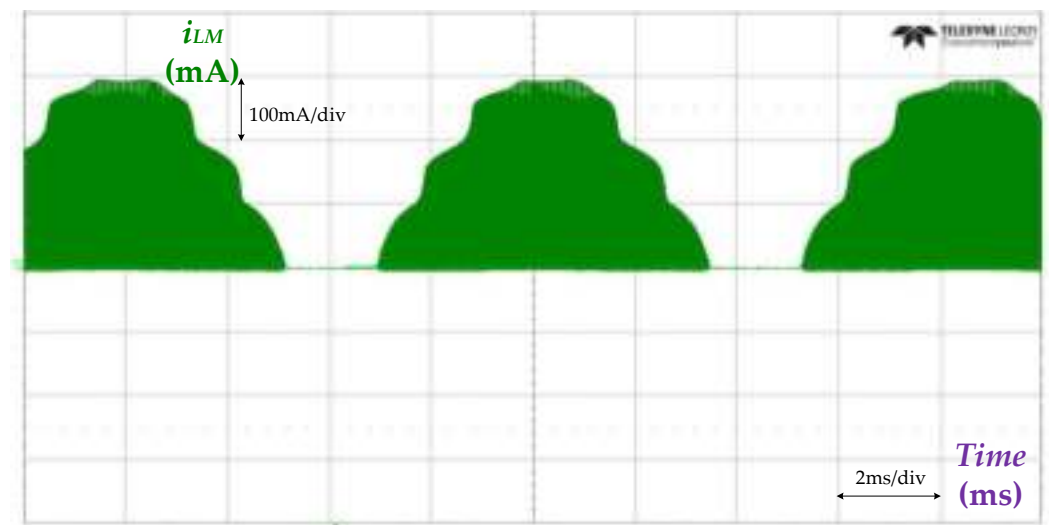


Figure 11. Measured magnetizing inductor current i_{LM} (100 mA/div); time scale: 2 ms/div.

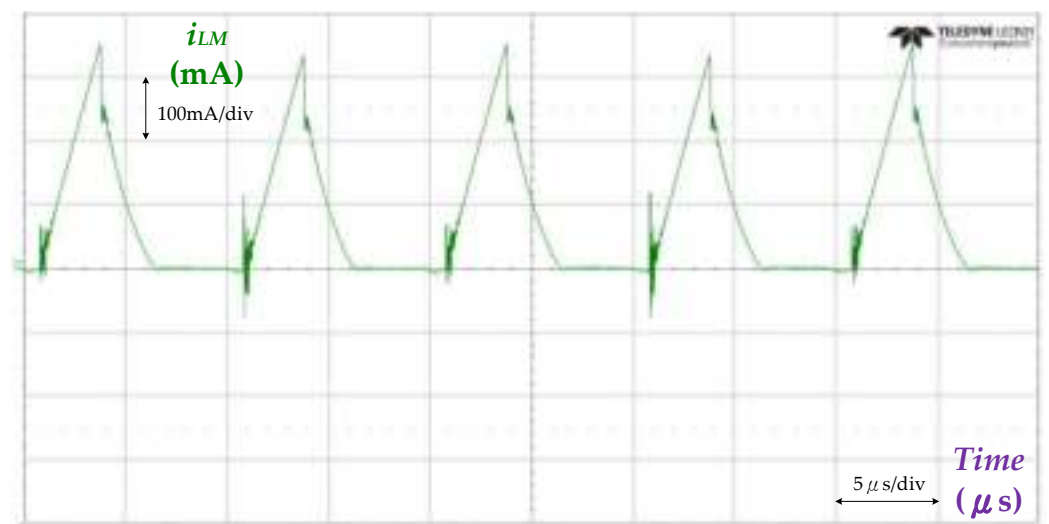


Figure 12. Measured unfolded waveform of the magnetizing inductor current i_{LM} (100 mA/div); time scale: 5 μ s/div.

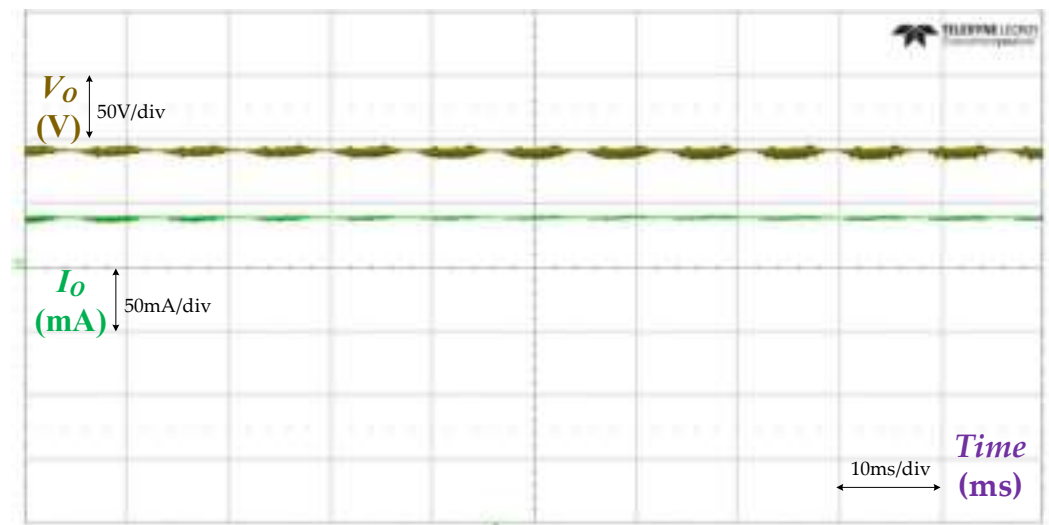


Figure 13. Measured waveforms of the output voltage V_o (50 V/div) and the output current I_o (50 mA/div); time scale: 10 ms/div.

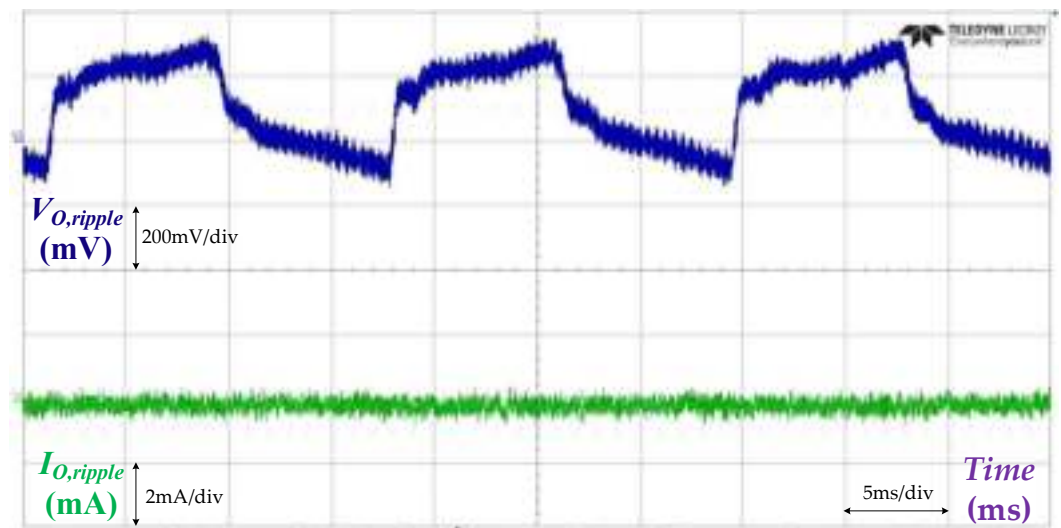


Figure 14. Measured waveforms of the output voltage ripple $V_{o,ripple}$ (200 mV/div) and the output current ripple $I_{o,ripple}$ (2 mA/div); time scale: 5 ms/div.

Table 4 shows the output voltage ripple factor and output current ripple factor of the prototype electronic lighting driver circuit for a deep ultraviolet LED disinfection and sterilization lamp at 110 volts RMS AC input voltage. As can be seen in Table 4, dividing the peak-to-peak output voltage of 143.8 millivolts by the average value of 91.35 volts yields a voltage ripple factor of 0.16%; dividing the peak-to-peak output current of 1.1658 milliamps by the average value of 38.57 milliamps yields a current ripple factor of 3.02%.

Figure 15 shows the measured waveforms of AC input voltage v_{AC} and input current i_{AC} . From the figure, it can be indicated that the AC input current follows the input voltage and the phases of the two waveforms are the same; therefore, the prototype of the proposed single-stage driver circuit for supplying the deep ultraviolet LED disinfecting and sterilizing lamp has the effect of power factor correction. The power factor and the total harmonic distortion factor of the input current were measured to be 0.9236 and 17.401%, respectively, using a power analyzer. In addition, the measured input power of the prototype drive circuit is 3.901 W and the output power is 3.583 W, and the efficiency of the prototype circuit is 91.85%.

Table 4. Measurement of output voltage ripple factor and output current ripple factor of electronic lighting prototype driver circuit utilizing SiC Schottky diodes for a deep ultraviolet LED disinfection and sterilization lamp.

Parameters	Values
Peak-to-peak value of the output voltage	143.8 mV
Mean value of the output voltage	91.35 V
Ripple factor of the output voltage	0.16%
Peak-to-peak value of the output current	1.1658 mA
Mean value of the output current	38.57 mA
Ripple factor of the output current	3.02%

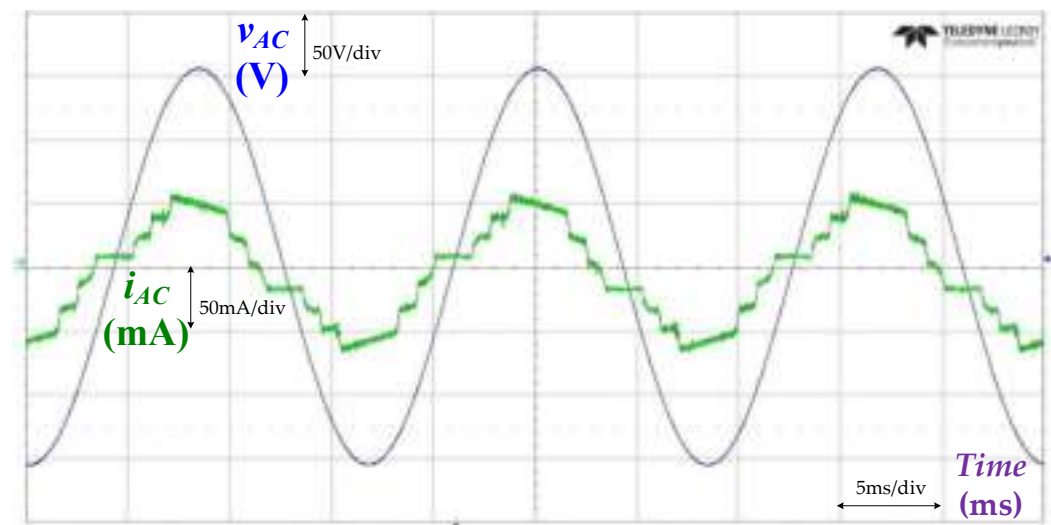


Figure 15. Measured input utility-line voltage v_{AC} (50 V/div) and current i_{AC} (50 mA/div); time scale: 5 ms/div.

Table 5 shows a comparison between the existing AC-DC LED driver in [19] which integrates an inverse buck-boost converter with a lossless snubber and supplies an 18 W-rated (60 V/0.3 A) power, and the proposed one which integrates a buck converter and a flyback converter and supplies a 3.6 W-rated (90 V/0.04 A) power. As can be seen from Table 3, both LED drivers are supplied by an input AC voltage of 110 V and use a single power switch. The proposed AC-DC LED driver circuit saves a capacitor, a magnetic element and a diode compared to the ones in [19]. In addition, the current THD and circuit efficiency of the proposed AC-DC LED driver is better than that of the existing driver.

Table 5. Comparisons between the existing AC-DC LED driver in [19] and the proposed one.

Item	Existing AC-DC LED Driver in Reference [19]	Proposed AC-DC LED Driver
Circuit topology	Integration of an inverse buck-boost converter with a lossless snubber	Integration of a buck converter and a flyback converter
Input AC voltage	110 V	110 V
Output power	18 W (60 V/0.3 A)	3.6 W (90 V/0.04 A)
Number of required switch	1	1
Number of required capacitors	4	3
Number of required magnetic element	3	2
Number of required diodes	7	6
Measured power factor	0.9737	0.9236
Measured current THD	18.422%	17.401%
Measured circuit efficiency	85.01%	91.85%

4. Conclusions

This paper proposes a single-stage electronic lighting prototype driver circuit utilizing SiC Schottky diodes for a deep ultraviolet LED disinfection and sterilization lamp, which integrates a buck converter and a flyback converter to form a single-stage, single-power-switch AC-DC power converter circuit architecture with a factor correction function. By designing the inductor in the AC-DC power converter to operate in discontinuous conduction mode, the effect of power factor correction can be achieved naturally. In addition, the proposed single-stage electronic lighting driver circuit employs a majority-carrier Schottky diode made of silicon carbide wide-bandgap semiconductor material and recovers the energy from the leakage inductance of the transformer, thereby improving the conversion efficiency of the driver circuit. In this paper, a 3.6 watt deep ultraviolet LED electronic lighting prototype driver circuit for a disinfection and sterilization lamp has been developed and tested. At an AC input voltage of 110 volts RMS and at rated output power, the results were: output voltage ripple factor less than 1%, output current ripple factor less than 4%, power factor greater than 0.9, input current total harmonic distortion factor less than 18%, and circuit efficiency greater than 90%. The single-stage deep ultraviolet LED electronic lighting prototype driver circuit proposed in this paper can simplify the number of power switches required in the driver circuit and has high circuit efficiency.

Author Contributions: C.-A.C. developed the circuit topology of the electronic lighting driver and designed the methodology; C.-H.C., H.-L.C. and E.-C.C. arranged the software resources and set up simulation along with performed circuit simulations; L.-F.L., S.-H.H. and C.-K.L. implemented the prototype electronic lighting driver circuit, and carried out the measurements of the circuit as well as analyzed experimental results with the guidance from C.-A.C.; C.-A.C. prepared the first draft of the article and revised the manuscript for submission. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science and Technology Council (NSTC) of Taiwan for its grant numbers MOST 110-2221-E-214-014, NSTC 111-2221-E-214-011 and NSTC 112-2221-E-214-005.

Data Availability Statement: The data presented in this study are available in the article.

Acknowledgments: The authors would like to express their sincerest gratitude to the National Science and Technology Council (NSTC) of Taiwan for its grant numbers MOST 110-2221-E-214-014, NSTC 111-2221-E-214-011 and NSTC 112-2221-E-214-005.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kebbi, Y.; Muhammad, A.I.; Sant'Ana, A.S.; Prado-Silva, L.D.; Liu, D.; Ding, T. Recent advances on the application of UV-LED technology for microbial inactivation: Progress and mechanism. *Compr. Rev. Food Sci. Food Saf.* **2020**, *19*, 3501–3527. [[CrossRef](#)] [[PubMed](#)]
2. Shatalov, M.; Jain, R.; Saxena, T.; Dobrinsky, A.; Shur, M. Development of deep UV LEDs and current problems in material and device technology. In *Semiconductors and Semimetals*; Elsevier: Amsterdam, The Netherlands, 2017; Volume 96, pp. 45–83.
3. Koutchma, T.; Popović, V.; Green, A. Overview of ultraviolet (UV) LEDs technology for applications in food production. In *Ultraviolet LED Technology for Food Applications*; Academic Press: Cambridge, MA, USA, 2019; pp. 1–23.
4. Akgün, M.P.; Ünlütürk, S. Effects of ultraviolet light emitting diodes (LEDs) on microbial and enzyme inactivation of apple juice. *Int. J. Food Microbiol.* **2017**, *260*, 65–74. [[CrossRef](#)] [[PubMed](#)]
5. Kaya, Z.; Yıldız, S.; Ünlütürk, S. Effect of UV-C irradiation and heat treatment on the shelf life stability of a lemon–melon juice blend: Multivariate statistical approach. *Innov. Food Sci. Emerg. Technol.* **2015**, *29*, 230–239. [[CrossRef](#)]
6. Guan, Z.; Liu, P.; Zhou, T.; Zhou, L.; Zhang, D.; Xie, Q.; Yu, Q.; He, Y.; Wang, S.; Wang, X.; et al. Study on the Light Field Regulation of UVC-LED Disinfection for Cold Chain Transportation. *Appl. Sci.* **2022**, *12*, 1285. [[CrossRef](#)]
7. Liang, J.J.; Liao, C.C.; Chang, C.S.; Lee, C.Y.; Chen, S.Y.; Huang, S.B.; Yeh, Y.F.; Singh, K.J.; Kuo, H.C.; Lin, Y.L.; et al. The effectiveness of far-ultraviolet (UVC) light prototype devices with different wavelengths on disinfecting SARS-CoV-2. *Appl. Sci.* **2021**, *11*, 10661. [[CrossRef](#)]
8. Juarez-Leon, F.A.; Soriano-Sanchez, A.G.; Rodriguez-Licea, M.A.; Perez-Pinal, F.J. Design and Implementation of a Germicidal UVC-LED Lamp. *IEEE Access* **2020**, *8*, 196951–196962. [[CrossRef](#)] [[PubMed](#)]

9. Alonso, J.M.; Costa, M.A.D.; Ordiz, C. Integrated Buck-Flyback Converter as a High-Power-Factor Off-Line Power Supply. *IEEE Trans. Ind. Electron.* **2008**, *55*, 1090–1100. [[CrossRef](#)]
10. Gacio, D.; Alonso, J.M.; Calleja, A.J.; Garcia, J.; Rico-Secades, M. A universal-input single-stage high-power-factor power supply for HB-LEDs based on integrated buck–flyback converter. *IEEE Trans. Ind. Electron.* **2010**, *58*, 589–599. [[CrossRef](#)]
11. Abdelmessih, G.Z.; Alonso, J.M.; Costa, M.A.D. Loss Analysis for Efficiency Improvement of the Integrated Buck–Flyback LED Driver. *IEEE Trans. Ind. Appl.* **2018**, *54*, 6543–6553. [[CrossRef](#)]
12. Abdelmessih, G.Z.; Alonso, J.M.; Tsai, W.T. Analysis and experimentation on a new high power factor off-line LED driver based on interleaved integrated buck flyback converter. *IEEE Trans. Ind. Appl.* **2019**, *55*, 4359–4369. [[CrossRef](#)]
13. Luz, P.C.; Bolán, P.E.; Maboni, T.; Cosetin, M.R.; da Silva, M.F.; do Prado, R.N. An integrated insulated Buck-Boost-Flyback converter to feed LED's lamps to street lighting with reduced capacitances. In Proceedings of the 2014 11th IEEE/IAS International Conference on Industry Applications, Juiz de Fora, Brazil, 7–10 December 2014; pp. 1–6.
14. Divya, K.M.; Parackal, R. High power factor integrated buck-boost flyback converter driving multiple outputs. In Proceedings of the 2015 Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, India, 27 November 2015; pp. 1–5.
15. Cheng, C.A.; Chung, T.Y. A Novel Integrated Driver with PFC for Supplying Light-Emitting Diodes. In Proceedings of the 41th Annual Conference of the IEEE Industrial Electronics Society (IECON 2015), Yokohama, Japan, 9–12 November 2015; pp. 3514–3518.
16. Poorali, B.; Adib, E. Analysis of the integrated SEPIC-flyback converter as a single-stage single-switch power-factor-correction LED driver. *IEEE Trans. Ind. Electron.* **2016**, *63*, 3562–3570. [[CrossRef](#)]
17. Cheng, C.-A.; Chang, C.-H.; Cheng, H.-L.; Chang, M.-T. A Novel LED Driver with Power Factor Correction Suitable for Streetlight Applications. In Proceedings of the International Future Energy Electronics Conference 2017 (IFEEC 2017)—ECCE Asia, Kaohsiung, Taiwan, 3–7 June 2017; pp. 1–4.
18. Cheng, C.A.; Chang, C.H.; Cheng, H.L.; Chang, E.C.; Lai, C.C. A novel LED tube lamp driver with power-factor correction for indoor lighting applications. In Proceedings of the 2018 IEEE 7th Global Conference on Consumer Electronics (GCCE), Nara, Japan, 9–12 October 2018; pp. 700–701.
19. Cheng, C.A.; Chen, Y.L.; Chen, Y.J.; Lai, S.W. A novel cost-effective AC-DC power supply for indoor LED lighting applications. In Proceedings of the 2019 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), Macao, China, 1–4 December 2019; pp. 1–3.
20. Cheng, C.A.; Lan, L.F.; Hou, S.H.; Lin, C.K. A Single-Stage Electronic Lighting Driver for a Deep Ultraviolet LED Disinfection and Sterilization Lamp. In Proceedings of the 2023 IEEE Workshop on Wide Bandgap Power Devices and Applications in Asia (WiPDA Asia), Hsinchu, Taiwan, 27–29 August 2023; pp. 1–3.
21. Effective Use of Power Devices—Silicon Carbide Power Devices Understanding & Application Examples Utilizing the Merits. ROHM Semiconductor, 2017. Available online: https://pages.rohm.com.tw/rs/247-PYD-578/images/TechWeb_E_DL_SiC.pdf (accessed on 31 July 2024).
22. SiC Power Devices and Modules Application Note, ROHM Semiconductor, 2020. Available online: https://fscdn.rohm.com/en/products/databook/applinote/discrete/sic/common/sic_appli-e.pdf (accessed on 31 July 2024).
23. AN966 APPLICATION NOTE, STMicroelectronics, “L6561, ENHANCED TRANSITION MODE POWER FACTOR CORRECTOR”, pp. 1–23, March 2003. Available online: https://www.st.com/resource/en/application_note/cd00004002-l6561-enhanced-transition-mode-power-factor-corrector-stmicroelectronics.pdf (accessed on 31 July 2024).
24. SEOULVIOSSYS. UV CA3535 Series (CUD5GF1B). Available online: https://media.digikey.com/pdf/Data%20Sheets/Seoul%20Semiconductor/CUD5GF1B_R0.1_Preliminary_2-3-20.pdf (accessed on 31 July 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Efficient Speech Signal Dimensionality Reduction Using Complex-Valued Techniques

Sungkyun Ko ¹  and Minho Park ^{2,*} ¹ Department of AI IT Convergence, Soongsil University, Seoul 06978, Republic of Korea; kosungkyun@naver.com² School of Electronic Engineering, Soongsil University, Seoul 06978, Republic of Korea

* Correspondence: mhp@ssu.ac.kr

Abstract: In this study, we propose the CVMFCC-DR (Complex-Valued Mel-Frequency Cepstral Coefficients Dimensionality Reduction) algorithm as an efficient method for reducing the dimensionality of speech signals. By utilizing the complex-valued MFCC technique, which considers both real and imaginary components, our algorithm enables dimensionality reduction without information loss while decreasing computational costs. The efficacy of the proposed algorithm is validated through experiments which demonstrate its effectiveness in building a speech recognition model using a complex-valued neural network. Additionally, a complex-valued softmax interpretation method for complex numbers is introduced. The experimental results indicate that the approach yields enhanced performance compared to traditional MFCC-based techniques, thereby highlighting its potential in the field of speech recognition.

Keywords: mel-frequency cepstral coefficients; complex-valued neural network error correction; backpropagation; complex-valued softmax



Citation: Ko, S.; Park, M. Efficient Speech Signal Dimensionality Reduction Using Complex-Valued Techniques. *Electronics* **2024**, *13*, 3046. <https://doi.org/10.3390/electronics13153046>

Academic Editors: Xin Geng, Padma Iyengar and Elke Pulvermüller

Received: 3 June 2024

Revised: 31 July 2024

Accepted: 31 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Today, speech signal processing and recognition play crucial roles in various application domains. These technologies are utilized for diverse purposes, such as converting speech commands into text or identifying the owner of a voice. The advancement of speech recognition technology can improve human–machine interaction, thereby enhancing productivity and efficiency across various industries. However, speech signals possess highly diverse forms and high-dimensional characteristics, making effective processing and recognition a persistent challenge. The diversity and high dimensionality of speech signals make dimensionality reduction and pattern recognition difficult, leading to increased computational costs. Therefore, efficiently reducing the dimensionality of speech signals and improving the performance of neural network-based speech recognition models is a significant challenge.

Speech processing and recognition technologies are essential in various applications, such as voice interfaces, virtual assistants, and automatic translation. Achieving more accurate and efficient speech signal processing and recognition results can significantly enhance the user experience of related services. Consequently, continuous research is being conducted in the field of speech signal processing and recognition. Speech signals exhibit high-dimensional characteristics, making effective dimensionality reduction challenging. Moreover, accurate recognition of diverse speech patterns requires intricate neural network models, which, in turn, increases computational costs [1]. Therefore, effectively reducing the dimensionality of speech signals without information loss and simultaneously enhancing the accuracy and efficiency of neural network-based speech recognition models is a crucial task.

Despite the rich literature on complex-valued networks, there have been few examples of their application in audio or speech processing. Given that the human auditory system

processes sound in the frequency domain, there is also meaning in exploring complex-valued neural networks for audio processing rather than relying solely on real-valued neural networks in the time domain [2].

Recent research has made significant progress in speech recognition by combining traditional methods like mel-frequency cepstral coefficients (MFCCs) with neural networks [3]. While the MFCC technique has been widely used for speech feature extraction, it has the drawback of being unable to perform lossless dimensionality reduction, as it uses only real values. In other words, reducing the number of MFCCs leads to information loss, which decreases speech recognition accuracy. These methods still exhibit limited effectiveness, and improved approaches beyond traditional learning methods are necessary.

Utilizing the complex representation of MFCC allows for the consideration of both real and imaginary parts, enabling lossless dimensionality reduction. Additionally, employing complex-valued neural networks can reflect phase changes through complex multiplication operations [4], better representing the characteristics of speech signals. This approach is expected to enable more efficient dimensionality reduction and neural network-based speech recognition compared to existing methods.

In this study, we propose the CVMFCC-DR algorithm for efficient and lossless speech signal dimensionality reduction. By combining CVMFCC-DR with complex-valued neural networks, this technique can demonstrate superior speech recognition performance compared to existing methods.

The main contributions of this study are as follows:

- Proposing an efficient dimensionality reduction technique (CVMFCC-DR) utilizing the complex representation of MFCCs.
- Enhancing speech recognition performance by combining complex-valued MFCCs and complex-valued neural networks.
- Proposing a new interpretation method for a complex-valued softmax function with complex inputs.

This paper begins by introducing the existing MFCC technique and complex-valued neural networks. It then provides a detailed explanation of the proposed CVMFCC-DR algorithm and the learning method for complex-valued neural networks [5]. Subsequently, the effectiveness of the proposed method is experimentally validated, and the results are analyzed. Finally, the paper concludes with a summary of the study's findings and suggests directions for future research.

2. Related Works

In the field of speech recognition, various neural network-based approaches and techniques are utilized. Different types of neural networks, such as CNNs, RNNs, LSTM, and DNNs, are used, each with their own advantages and disadvantages [3]. Moreover, hybrid models that combine traditional methods with neural network structures are also being researched. Hybrid models can optimize performance by combining the strengths of each structure, but they also come with the drawback of increased complexity in implementation and interpretation [6].

Amidst these developments, traditional feature extraction techniques like the MFCC technique are still widely used as inputs for neural networks [3]. Additionally, exploring novel neural architectures such as complex-valued neural networks could uncover new perspectives and potential breakthroughs in speech recognition capabilities. Recently, there have been studies showing that FPGAs can be used to significantly improve complex number computation speed [7]. These research results empirically demonstrate that the hardware implementation of complex number operations can bring about practical performance improvements. Furthermore, a recent study implemented complex-valued neural networks on FPGAs for classification, demonstrating superior accuracy compared to real-valued neural networks and achieving significant speed improvements over CPU-based systems [8]. This gives rise to the expectation that in the future, CPUs that currently only support real number operations may one day support complex number operations as well.

Based on these studies, utilizing complex number representations enables the encoding of real number pairs as single complex numbers, effectively reducing the input dimension by half. This reduction in dimensionality decreases the number of model parameters and enhances computational efficiency. Consequently, experimental evidence shows that complex-valued neural networks can achieve equal or superior performance with fewer parameters compared to conventional real-valued neural networks.

As research continues to push the boundaries, a synergistic combination of proven methods and innovative approaches may pave the way for the next generation of robust and high-performing speech recognition systems.

3. Background Knowledge

Various studies are being conducted in the field of speech recognition and classification, and in this section, we introduce MFCCs (mel-frequency cepstral coefficients) [3]. The MFCC technique is one of the feature extraction methods commonly used in the fields of speech signal processing and speech recognition. The MFCC technique represents the frequency characteristics of speech signals and is primarily utilized to effectively represent and recognize speech signals. The basic principle of the MFCC technique is illustrated in Figure 1.



Figure 1. An overview of the process of the MFCC technique.

Pre-emphasis: Pre-emphasis is essentially a high-pass filter. Due to the structure of the human body, high-frequency components tend to decrease in the actual emitted sound during speech production. Therefore, pre-emphasis is the process of applying a high-pass filter to amplify the attenuated high-frequency components first.

Framing and Windowing: Speech signals are divided into short segments called frames for analysis. This is because speech characteristics remain relatively stable over brief periods. These frames slightly overlap to capture speech changes more effectively. Each frame is processed using a window technique, which enhances analysis. Common window types include Hamming and Hanning windows. This framing method allows for more accurate analysis of complex speech data, which is crucial for developing speech recognition and synthesis technologies. While specific implementation details may vary, these fundamental principles are widely used in speech processing systems.

Discrete Fourier Transform: For calculating the power spectrum, DFT (Discrete Fourier Transform) is used to compute the power spectrum for each frame obtained earlier. This process enables the discernment of the frequency-specific energy distribution of the speech signal across each frame.

Mel-Frequency Filter Bank: The mel band-pass filter is a set of filters designed to extract speech signals in a nonlinear manner, mimicking the human auditory system. Typically constructed with 40 triangular filters, it analyzes the frequency characteristics of speech signals.

Discrete Cosine Transform: DCT (Discrete Cosine Transform) expresses a finite sequence of data points as a sum of cosine functions oscillating at different frequencies. In the MFCC process, DCT is applied to the mel filter bank to select acceleration coefficients or to separate the relationship in log-spectral magnitudes from the filter bank.

Speech recognition using MFCCs can be effectively employed in various environments [9], and this method can achieve high accuracy when combined with deep learning models. Therefore, MFCC serves as a crucial input for deep learning models for speech recognition [10].

4. Proposed Method

In this section, we propose a method for efficiently performing speech signal dimensionality reduction using complex-valued MFCCs and complex-valued neural networks. An overview of the proposed approach is presented in Figure 2. This figure illustrates the proposed approach for voice classification using complex-valued neural networks. The process commences with voice input, followed by feature extraction using mel-frequency cepstral coefficients (MFCCs). These features are then subjected to dimensionality reduction and transformed into complex-valued representations, resulting in complex-valued MFCCs. The processed features are subsequently inputted into a complex-valued neural network. The network's performance is enhanced through the selection of appropriate activation functions and learning algorithms. Ultimately, the neural network's output is utilized for classification purposes.

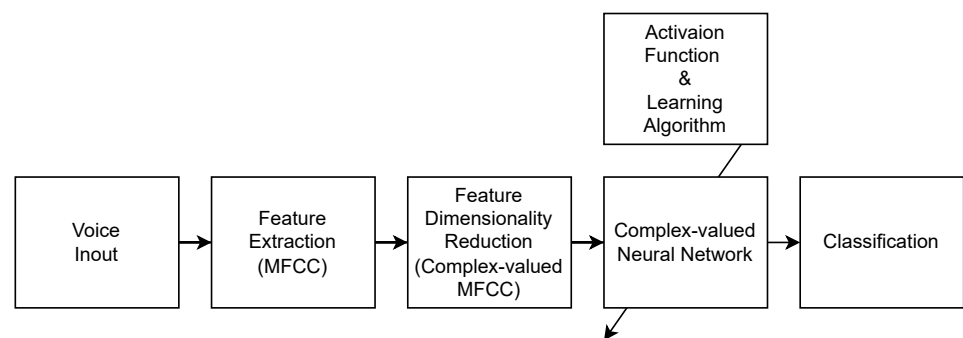


Figure 2. Overview of the proposed approach.

Complex-valued MFCCs: The number of MFCCs used in speech recognition can vary depending on specific applications and models. However, typically, MFCCs ranging from 13 to 40 are used. This range is known to effectively operate in various speech recognition systems. Generally, a higher number of MFCCs can allow the model to capture more frequency and spectral detail, but it also increases the computational cost of the model. Therefore, the selected number of MFCCs is determined considering a balance between computational efficiency and model performance.

When representing physical problems using complex numbers, there are two main methods. One is a simple substitution method, and the other is a more elegant approach known as mathematical equivalence, like the Fourier Transform. Substitution takes two real physical parameters and places one in the real part of the complex number and one in the imaginary part. This allows the two values to be manipulated as a single entity, i.e., a single complex number [11]. The method proposed for dimensionality reduction in this paper involves converting n MFCC values into complex numbers. The first MFCC is used as the real part, and the second MFCC is used as the imaginary part of the complex number. Subsequently, each MFCC alternates between being the real and imaginary parts within the complex number. This pattern continues up to the n th MFCC, resulting in $n/2$ complex numbers. For example, when we have n MFCC values, they could be as follows:

$$[x_1, x_2, x_3, x_4, \dots, x_{n-1}, x_n]$$

In this case, the first MFCC x_1 is used as the real part of the first complex number. The second MFCC x_2 is used as the imaginary part of the first complex number. The third MFCC x_3 is used as the real part of the second complex number. The fourth MFCC x_4 is used as the imaginary part of the second complex number. This pattern continues alternating between real and imaginary parts up to the n th MFCC x_n , which becomes the imaginary part of the $(n/2)$ -th complex number. Therefore, with n MFCC values, $n/2$ complex numbers are generated. The first complex number is $x_1 + jx_2$, the second complex

number is $x_3 + jx_4$, and so on until the last complex number, which is $x_{n-1} + jx_n$. We have illustrated this explanation in Figure 3.

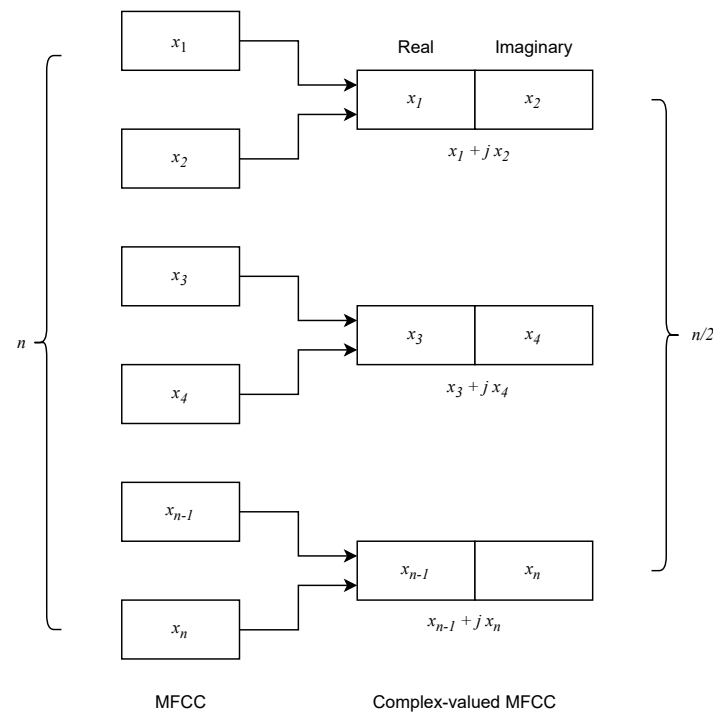


Figure 3. Process of generating complex-valued MFCCs.

Complex-Valued Neural Network: Complex-valued neural networks are neural network models that use complex numbers as input values or weights, and they have the advantage of handling richer information compared to real-valued neural networks. Furthermore, various research studies have been conducted in various fields, and complex-valued dimensionality reduction methods or classification and detection using complex numbers are being effectively utilized. For this purpose, the multiplication of complex inputs and complex weights is performed, which implies rotation in the complex plane and thus introduces changes in phase [5]. The sum of the multiplication of input X and weight W can be represented as Equations (1) and (2).

$$Z = W_0 + W_1X_1 + W_2X_2 + \dots + W_nX_n \quad (1)$$

$$Z = \sum_n W_n \cdot X_n \quad (2)$$

Activation Function and Learning Algorithm: In complex-valued neural networks, a phase-dependent activation function is used, forcing the magnitude information to be normalized to a unit circle of 1 and only the extraction of the phase information, as shown in Equation (3).

$$P(Z) = \frac{Z}{|Z|} = e^{j \cdot \arg(Z)} \quad (3)$$

The learning algorithm of complex-valued neural networks is as follows. When there is a desired target t and an output value z , as shown in Figure 4, the weights are adjusted according to Equation (4) to gradually move in the direction of the desired target [12].

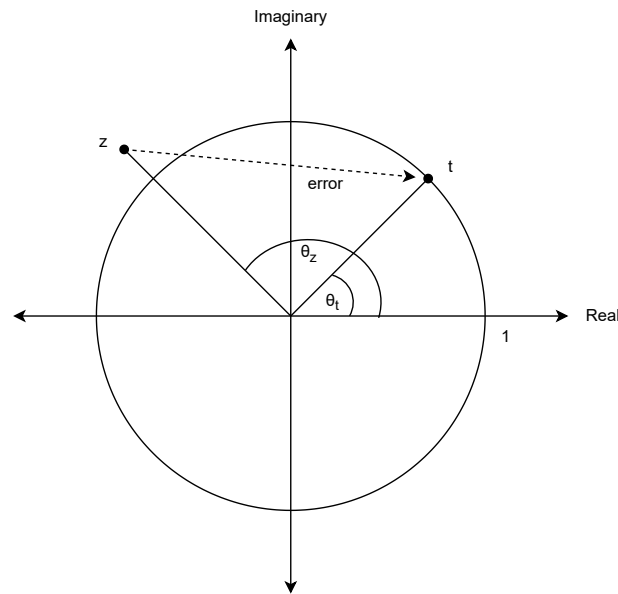


Figure 4. Learning algorithm for CVNN.

$$t = \sum_n (W_n + \Delta W_n) \cdot X_n \quad (4)$$

The error with respect to the training target t is $e = t - z$, as shown in Equation (5).

$$e = \sum_n (W_n + \Delta W_n) \cdot X_n - \sum_n (W_n) \cdot X_n = \sum_n (\Delta W_n) \cdot X_n \quad (5)$$

To simplify and expedite the calculation of the error with respect to weight changes, assuming that each node contributes equally to the error, the summation symbol σ is removed from the equation, and both sides are divided by the number of contributing nodes N in the complex-valued neural network for computational convenience. The resulting equation is Equation (6).

$$\Delta W_n \cdot X_n = \frac{e}{N} \quad (6)$$

To obtain ΔW_n in order to change the weights, multiply both sides of Equation (6) by X_n^{-1} . X_n^{-1} becomes $\overline{X_n}$ when multiplied by the complex conjugate $\overline{X_n}$, with n being both the numerator and denominator; thus, $X_n^{-1} = \frac{\overline{X_n}}{|X_n|^2}$, where $|X_n| = 1$ for X_n within the unit circle. Therefore, $X_n^{-1} = \overline{X_n}$, and Equation (6) is derived again as Equations (7) and (8).

$$\Delta W_n \cdot X_n \cdot X_n^{-1} = \frac{e}{N} \cdot X_n^{-1} = \frac{e}{N} \cdot \overline{X_n} \quad (7)$$

$$\Delta W_n = \frac{e}{N} \cdot \overline{X_n} \quad (8)$$

In this way, the obtained change in weights can be observed in Equation (8). This change represents the adjustment of complex weights for each node contributing to the error with respect to the training target and is uniformly applied across the entire network. Through this process, complex-valued neural networks adjust closer to the target utilizing phase-dependent activation functions and efficient learning algorithms while handling rich information through complex multiplication operations between inputs and weights [13].

Complex Domain Backpropagation: When using hidden layers, the error from the output layer is backpropagated to the hidden layers to calculate how much each node

in the hidden layers contributed to that error. Based on this, the weights between the hidden and output layers need to be adjusted appropriately. This process is called error backpropagation, through which the neural network can adjust the weights to minimize the error in the output values during learning. The structure of a multi-layer complex-valued neural network is illustrated in Figure 5.

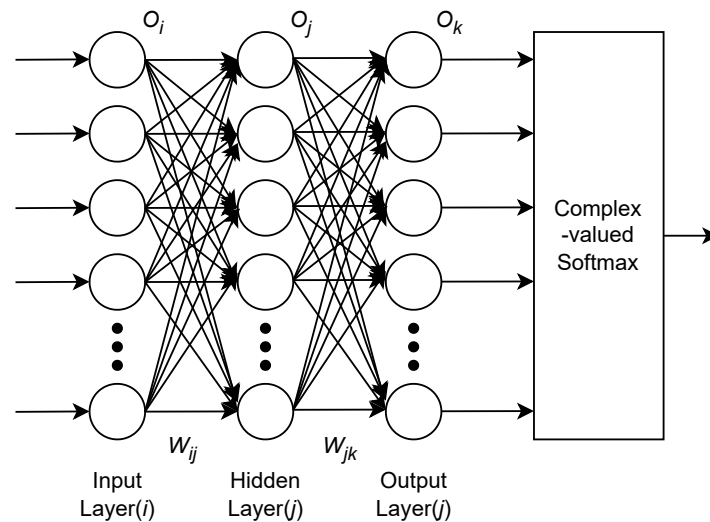


Figure 5. Multi-layer CVNN.

In this paper, an error correction method is used for the output layer k to update the weights, while an error backpropagation method is employed for the hidden layer j to examine its influence on the output. In other words, the output layer k utilizes the error correction method to avoid falling into local minima, while the error backpropagation method is applied to the hidden layer j to ensure better distribution of errors from the output layer to the hidden layer. This combination of error backpropagation in the hidden layer and error correction in the output layer can be expected to improve the problem of the entire neural network getting stuck in local minima while enabling faster convergence. The error backpropagation method for complex-valued neural networks is as follows.

When using gradient descent for weight updates in complex-valued neural networks, Wirtinger calculus is employed for differentiating complex functions. As complex functions comprise real and imaginary parts, Wirtinger calculus conducts differentiation separately for the real and imaginary parts. For a complex number $z = x + jy$ defined with real and imaginary parts x and y , respectively, and a complex function $f(z) = u(x, y) + jv(x, y)$, the Wirtinger derivatives are defined as follows [14].

$$\frac{\partial f}{\partial z} = \frac{1}{2} \left(\frac{\partial f}{\partial x} - j \frac{\partial f}{\partial y} \right) = \frac{1}{2} \left(\frac{\partial u}{\partial x} + j \frac{\partial v}{\partial x} - j \frac{\partial u}{\partial y} + \frac{\partial v}{\partial y} \right) \quad (9)$$

$$\frac{\partial f}{\partial \bar{z}} = \frac{1}{2} \left(\frac{\partial f}{\partial x} + j \frac{\partial f}{\partial y} \right) = \frac{1}{2} \left(\frac{\partial u}{\partial x} + j \frac{\partial v}{\partial x} + j \frac{\partial u}{\partial y} - \frac{\partial v}{\partial y} \right) \quad (10)$$

$$df = \frac{\partial f}{\partial z} dz + \frac{\partial f}{\partial \bar{z}} d\bar{z} \quad (11)$$

In this context, $\bar{z}$ denotes the complex conjugate of z . In the Wirtinger derivative method, complex functions can be treated as extensions of real functions by dividing them into real and imaginary parts for differentiation. The following describes the updating process of complex variables using Step Size $\alpha/2$ and Loss Function L .

$$x_{n+1} = x_n - \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial x} \quad (12)$$

$$y_{n+1} = y_n - \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial y} \quad (13)$$

By substituting $z_{n+1} = x_{n+1} + j y_{n+1}$ into Equations (12) and (13) and then substituting the Wirtinger derivative at the end of the derivation process, it becomes as follows.

$$\begin{aligned} z_{n+1} &= x_n - \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial x} + j \left(y_n - \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial y} \right) \\ &= x_n - \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial x} + j y_n - j \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial y} \\ &= x_n + j y_n - \left(\frac{\alpha}{2}\right) \left(\frac{\partial L}{\partial x} + j \frac{\partial L}{\partial y} \right) \\ &= z_n - \alpha \frac{1}{2} \left(\frac{\partial L}{\partial x} + j \frac{\partial L}{\partial y} \right) \\ &= z_n - \alpha \frac{\partial L}{\partial \bar{z}} \end{aligned} \quad (14)$$

The partial derivative of L with respect to the complex conjugate $\frac{\partial L}{\partial \bar{z}}$ is referred to as the Conjugate Wirtinger Derivative. This demonstrates how the Conjugate Wirtinger Derivative can be utilized to simplify the update formula for complex variables. In terms of using the above method, rewriting Equation (5) as an output layer error yields Equation (15), where t_k represents the target value, and o_k represents the actual output value of output layer k .

$$e_k = t_k - o_k \quad (15)$$

And the Loss Function is defined as Equation (16).

$$L = e_k^2 = |t_k - o_k| \overline{|t_k - o_k|} \quad (16)$$

If the weights are large, the error is more likely to propagate back to the hidden layers. The hidden layer error e_j can be expressed by the following equation.

$$e_j = W_{jk}^T \cdot e_k \quad (17)$$

Using the result of Equation (14), the gradient of the error function with respect to the weight $\bar{W}_{ij}$ between input layer i and hidden layer j can be expressed as follows. Here, o_i is the output value of input layer i , and o_j is the output value of hidden layer j .

$$\frac{\partial L}{\partial \bar{W}_{ij}} = \frac{\partial L}{\partial o_j} \cdot \frac{\partial o_j}{\partial \bar{W}_{ij}} \quad (18)$$

$$\frac{\partial L}{\partial \bar{W}_{ij}} = -e_j \cdot \frac{\partial}{\partial \bar{W}_{ij}} P \left(\sum \bar{W}_{ij} \cdot \bar{o}_i \right) \quad (19)$$

$$\frac{\partial L}{\partial \bar{W}_{ij}} = -e_j \cdot P' \cdot \bar{o}_i \quad (20)$$

In Equations (19) and (20), P represents the phase-dependent activation function used in the hidden layer output, and P' is the derivative of the dependent activation function. The updating of hidden layer weights can be expressed as follows.

$$new\ W_{ij} = old\ W_{ij} - lr \cdot \frac{\partial L}{\partial \bar{W}_{ij}} \quad (21)$$

When dealing with complex variables, it is necessary to use $\frac{\partial L}{\partial \bar{W}_{ij}}$ instead of $\frac{\partial L}{\partial W_{ij}}$ for updating. Here, lr plays a role in adjusting the intensity of the change to prevent overshooting, and it is called the learning rate. Additionally, the derivative of the activation function P' is a complex function, and to find the derivative of a complex function, the Wirtinger derivative is used [15].

Complex-Valued Softmax: The softmax function, which is used as the activation function in the output layer, takes input values and normalizes each value to be between 0 and 1. It ensures that the sum of the normalized values is always 1. This function allows for the interpretation of results as probabilities. It can be expressed as Equation (22).

$$Softmax(Z)_i = \frac{e^{Z_i}}{\sum_{i=1}^n e^{Z_i}} \quad (22)$$

However, the softmax function can only interpret results as probabilities for real number inputs, and for complex number inputs, a different interpretation method is needed. In this paper, we propose a complex-valued softmax interpretation method for complex number inputs. The result of the softmax function for complex number inputs yields complex output values, and since the sum of all output values is $1 + 0j$, we propose an algorithm that calculates the distance between the output values and $1 + 0j$ in the complex plane and selects the index of the closest value. This can be expressed as Equation (24).

$$distance_i = |Softmax(Z)_i - \omega| \quad (23)$$

$$closest\_index = \operatorname{argmin}(distance_i) \quad (24)$$

Here, $\omega = 1 + 0j$ is defined. Consequently, we decide to classify the complex number corresponding to $closest\_index$ as class 1.

5. Performance Analysis

In our experiments of speech signal recognition, a detailed comparison between the results of real-valued neural networks and complex-valued neural networks was conducted. Performance analysis was carried out alongside the pre-processing of complex-valued mel-frequency cepstral coefficients (MFCCs) for dimensionality reduction. Additionally, the impact of complex-valued MFCCs on speech recognition results was observed. Specifically, in recognizing given speech signals using neural networks, the influence of MFCC values and their complexification effect on model performance was evaluated and compared to determine whether it is possible to enhance performance while reducing the number of parameters.

Experimental Method: Figure 6 depicts the architecture of a multi-layer complex-valued neural network used for training and testing with a dataset consisting of 8000 short speech samples, each containing eight different words and lasting less than one second. The speech data used in this study are part of the speech command dataset collected by Google and distributed under the CC BY license.

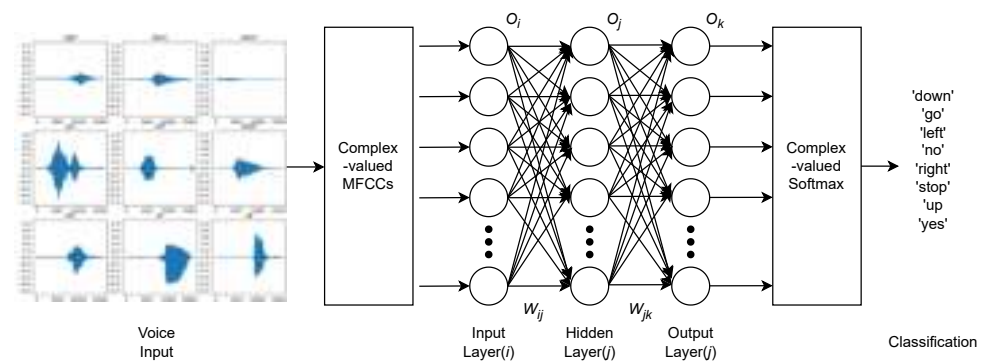


Figure 6. Experimental setup of multi-layer CVNN.

Complex-valued neural networks are structurally similar to real-valued neural networks, but the difference lies in the use of complex numbers, which introduces phase changes as the sum of products between inputs and weights rotates in the complex plane. These phase changes play an important role in complex-valued neural networks. The complex-valued neural network consists of an input layer with dimension-reduced complex-valued MFCCs obtained from the speech data through the complexification process shown in Figure 3, a hidden layer with 500 nodes, and an output layer with 8 nodes for classification. All nodes and weights use complex numbers. The final results were classified at the output layer. The dataset comprised 7000 samples for training and 1000 samples for testing. Eight-word audio clips were used as data, and experiments were conducted by dividing the input data based on whether complex-valued MFCC transformation was applied or not. In the case of using real-valued MFCCs, a real-valued neural network was employed, while a complex-valued neural network was used with complex-valued MFCCs. Both the real-valued and complex-valued neural networks were trained under the same conditions: $\text{lr} = 0.01$, batch size = 50, and number of epochs = 50. For the input data, the results of three different dimensionality reduction pre-processing steps were compared with the real-valued neural network. These pre-processing steps included MFCC dimensions of 20, 10, and 5. The length of the input signal for the neural network can be determined as shown in Equation (25). Specifically, the input speech signal was divided into 32 frames, and MFCCs were generated for each frame.

$$\text{The length of input signal data} = \text{MFCCs} \times \text{frames} \quad (25)$$

For instance, in the case of complex-valued MFCC 20, real-valued MFCC 40 is computed first. Using Equation (25), the length of the input signal for real-valued MFCC 40 becomes $40 \times 32 = 1280$. If we then convert these 40 MFCCs into complex numbers, as illustrated in Figure 3, dimensionality reduction is achieved to half the length, resulting in 16 frames. Consequently, the input signal length becomes $40 \times 16 = 640$, which is equivalent to the input signal length of real-valued MFCC 20. Similar dimensionality reduction can be performed for complex-valued MFCC 10 and complex-valued MFCC 5 using the same method. The total number of parameters for the entire neural network obtained through this process is presented in Table 1. In this table, $I(n)$ represents the number of neurons in the input layer, $H(n)$ represents the number of neurons in the hidden layer, and $O(n)$ represents the number of neurons in the output layer.

Table 1. Total number of parameters in neural network.

MFCC	Real-Valued MFCC	Complex-Valued MFCC
20	$I(20 \times 32 = 640) \times H(500) + H(500) \times O(8) = 324,000$	$I(40 \times 16 = 640) \times H(500) + H(500) \times O(8) = 324,000$
10	$I(10 \times 32 = 320) \times H(500) + H(500) \times O(8) = 164,000$	$I(20 \times 16 = 320) \times H(500) + H(500) \times O(8) = 164,000$
5	$I(5 \times 32 = 160) \times H(500) + H(500) \times O(8) = 84,000$	$I(10 \times 16 = 160) \times H(500) + H(500) \times O(8) = 84,000$

Complex-valued MFCC 20 vs. Real-valued MFCC 20: As shown in Figure 7, complex-valued MFCC 20 achieved a maximum training accuracy of 99.40% and a testing accuracy of 84.90% despite having the same input data length as real-valued MFCC 20, which achieved a maximum training accuracy of 98.13% and a testing accuracy of 83.10%. The results obtained using complex-valued MFCCs outperformed those obtained using real-valued MFCCs in terms of both training and testing accuracy.

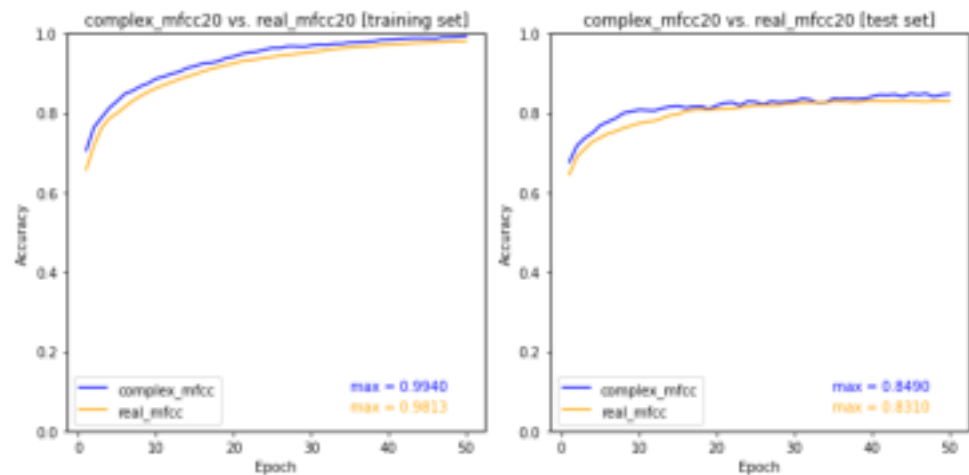


Figure 7. Comparison of complex-valued MFCC 20 and real-valued MFCC 20 results.

Complex-valued MFCC 10 vs. Real-valued MFCC 10: As shown in Figure 8, complex-valued MFCC 10 achieved a maximum training accuracy of 96.90% and a testing accuracy of 83.80% despite having the same input data length as real-valued MFCC 10, which achieved a maximum training accuracy of 94.24% and a testing accuracy of 81.20%. The results obtained using complex-valued MFCCs outperformed those obtained using real-valued MFCCs in terms of both training and testing accuracy.

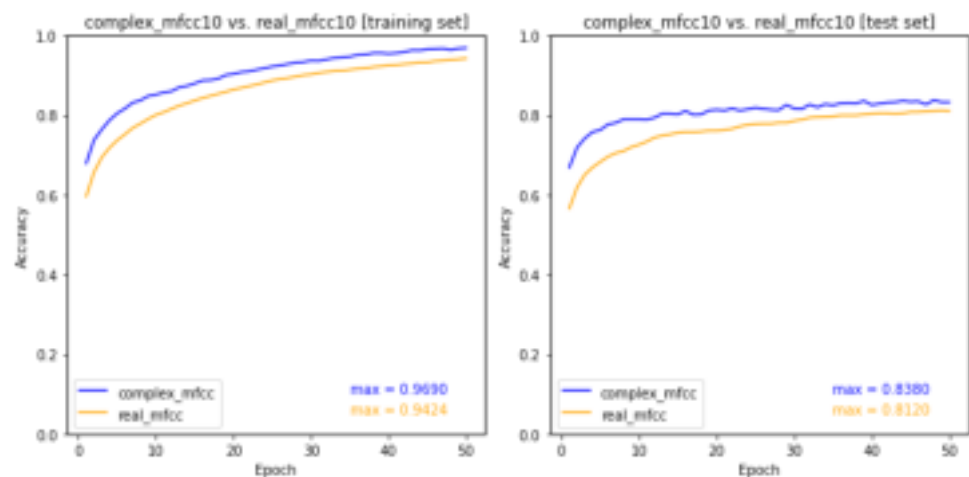


Figure 8. Comparison of complex-valued MFCC 10 and real-valued MFCC 10 results.

Complex-valued MFCC5 vs. Real-valued MFCC5: As shown in Figure 9, complex-valued MFCC 5 achieved a maximum training accuracy of 92.33% and a testing accuracy of 81.40% despite having the same input data length as real-valued MFCC 5, which achieved a maximum training accuracy of 86.34% and a testing accuracy of 74.90%. The results obtained using complex-valued MFCCs outperformed those obtained using real-valued MFCCs in terms of both training and testing accuracy.

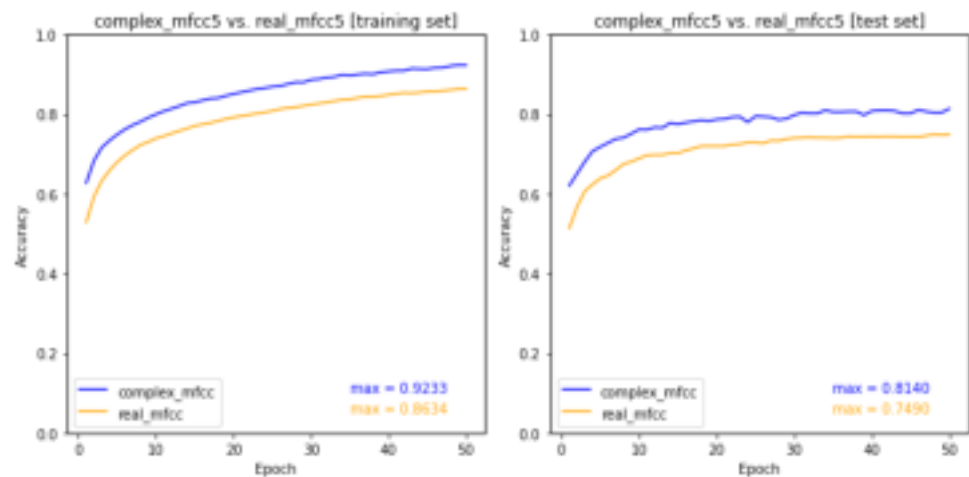


Figure 9. Comparison of complex-valued MFCC 5 and real-valued MFCC 5 results.

Overall Analysis Results: Upon reviewing the experimental results, it was observed that using complex-valued MFCCs yields higher performance compared to using real-valued MFCCs, even with the same input data length. Therefore, it was confirmed that employing complex-valued MFCCs and complex-valued neural networks leads to better results. The variation in accuracy according to the number of MFCC features is illustrated in Figure 10. Although not shown here, the same results were obtained using the F1 score.

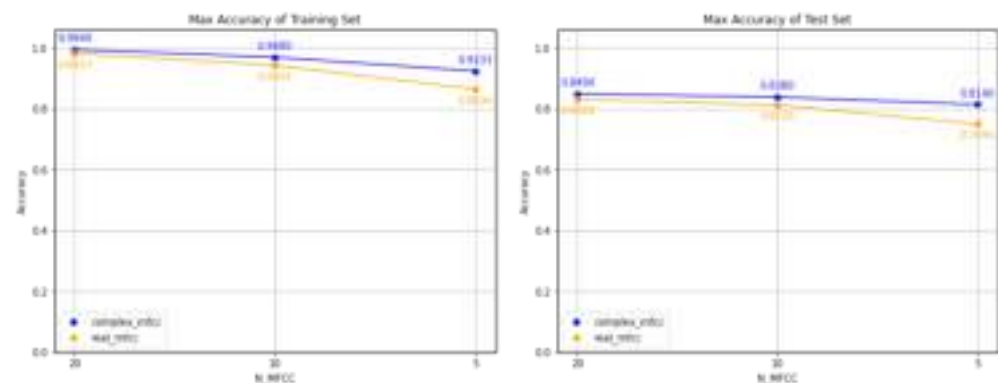


Figure 10. Accuracy variation with No. of MFCCs.

If there is a sufficient number of MFCCs, with increasing epochs, it appears that the performance using real numbers begins to converge towards that using complex numbers. However, Figure 9 shows a different outcome. Complex-valued MFCC 5 has 5 dimensionally reduced complex MFCCs obtained from 10 MFCCs through the complexification process shown in Figure 3. A real-valued MFCC 5 with only 5 MFCCs cannot catch up with complex-valued MFCC 5 even as the epochs increase. In other words, when the number of MFCCs is low, even as the epochs increase, the results of real-valued MFCCs do not match the performance of complex-valued MFCCs. As shown in Figure 10, despite having the same number of parameters, the results of complex-valued MFCC 5 are superior to those of real-valued MFCC 5. Therefore, it was observed that when using complex-valued MFCCs, the change in accuracy as the number of MFCC features decreases is smoother compared to when using real-valued MFCCs.

6. Conclusions

In this paper, we introduced complex-valued neural networks to address the problem of speech recognition. Our experiments utilized complex-valued MFCCs as inputs for a

complex-valued neural network. The results demonstrated superior performance when employing complex-valued MFCCs, with a more gradual change in accuracy as the number of MFCC features decreased, compared to using real-valued MFCCs. This suggests that complex-valued MFCCs can help to maintain performance with fewer features. These findings provide valuable insights into the potential of complex-valued neural networks and demonstrate the effectiveness of combining complex-valued MFCCs with complex-valued neural networks in speech recognition.

Despite these promising results, we acknowledge the certain limitations of our approach. This study was conducted on a limited dataset, which may not fully represent the diversity of real-world speech patterns. Moreover, the complex-valued neural network architecture described in this study is relatively simple and may not capture all the intricacies of speech signals.

Looking ahead, we aim to build upon this work to further enhance the accuracy and performance of speech signal processing and recognition models. Our future research will focus on designing more sophisticated complex-valued neural network models and conducting experiments on diverse datasets to improve performance. Additionally, we plan to explore the application of complex-valued neural networks to other fields beyond speech recognition, potentially expanding their use in various signal processing domains [16].

Author Contributions: Conceptualization, S.K.; project administration, M.P.; supervision, M.P.; writing—original draft, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was jointly supported by the National Research Foundation of Korea (NRF) via a grant provided by the Korea government (MSIT) (grant no. NRF-2023R1A2C1005461), and by the MSIT (Ministry of Science and ICT), Korea, under the Convergence Security Core Talent Training Business Support Program (IITP-2024-RS-2024-00426853) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation).

Data Availability Statement: The original contributions presented in the study are included in the article. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tebelskis, J. Speech Recognition Using Neural Networks. Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 1995.
2. Sarroff, A.M. Complex Neural Networks for Audio. Ph.D. Thesis, Dartmouth College, Hanover, NH, USA, 2018.
3. Abdul, Z.K.; Al-Talabani, A.K. Mel Frequency Cepstral Coefficient and Its Applications: A Review. *IEEE Access* **2022**, *10*, 122136–122158. [\[CrossRef\]](#)
4. Barrachina, J.A.; Ren, C.; Vieillard, G.; Morisseau, C.; Ovarlez, J.P. Theory and Implementation of Complex-Valued Neural Networks. *arXiv* **2023**, arXiv:2302.08286.
5. Aizenberg, I. *Complex-Valued Neural Networks with Multi-Valued Neurons*; Springer: Berlin/Heidelberg, Germany, 2016.
6. Graves, A.; Jaitly, N.; Mohamed, A.R. Hybrid speech recognition with Deep Bidirectional LSTM. In Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, 8–12 December 2013; pp. 273–278. [\[CrossRef\]](#)
7. Paz, P.; Garrido, M. Efficient Implementation of Complex Multipliers on FPGAs Using DSP Slices. *J. Signal Process. Syst.* **2023**, *95*, 543–550. [\[CrossRef\]](#)
8. Ahmad, M.; Zhang, L.; Chowdhury, M.E.H. FPGA Implementation of Complex-Valued Neural Network for Polar-Represented Image Classification. *Sensors* **2024**, *23*, 2627. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Anggraeni, D.; Sanjaya, W.S.; Nurasyidiek, M.Y.; Munawwaroh, M. The Implementation of Speech Recognition using Mel-Frequency Cepstrum Coefficients (MFCC) and Support Vector Machine (SVM) Method Based on Python to Control Robot Arm. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *288*, 012042. [\[CrossRef\]](#)
10. Dhanjal, A.S.; Singh, W. *A Comprehensive Survey on Automatic Speech Recognition Using Neural Networks*; Springer Nature: Berlin/Heidelberg, Germany, 2023.
11. Smith, S.W. *The Scientist and Engineer's Guide to Digital Signal Processing*, 2nd ed.; California Technical Publishing: San Diego, CA, USA, 1999; pp. 557–558.
12. Bassey, J.; Li, X.; Qian, L. A Survey of Complex-Valued Neural Networks. *arXiv* **2021**, arXiv:2101.12249v1.
13. MYONeuralNet. Complex-Valued Neural Networks—Experiments. 2016. Available online: <http://makeyourownneuralnetwork.blogspot.com/2016/05/complex-valued-neural-networks.html> (accessed on 19 February 2021).

14. Fischer, R. Wirtinger Calculus. In *Precoding and Signal Shaping for Digital Transmission*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2002; pp. 405–413.
15. PyTorch. Automatic Differentiation Package—torch.autograd, n.d. Available online: <https://pytorch.org/docs/stable/notes/autograd.html> (accessed on 3 November 2023).
16. Lee, C.; Hasegawa, H.; Gao, S. Complex-Valued Neural Networks: A Comprehensive Survey. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 1433–1454. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Job Recommendations: Benchmarking of Collaborative Filtering Methods for Classifieds

Robert Kwieciński ^{1,2} , Tomasz Górecki ^{2,*} , Agata Filipowska ³  and Viacheslav Dubrov ¹

¹ OLX Group, Królowej Jadwigi 43, 61-872 Poznań, Poland; r.kwiecinskipl@gmail.com (R.K.); slavadubrov@gmail.com (V.D.)

² Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Uniwersytetu Poznańskiego 4, 61-614 Poznań, Poland

³ Department of Information Systems, Poznań University of Economics and Business, Al. Niepodległości 10, 61-875 Poznań, Poland; agata.filipowska@ue.poznan.pl

* Correspondence: tomasz.gorecki@amu.edu.pl

Abstract: Classifieds pose numerous challenges for recommendation methods, including the temporary visibility of ads, the anonymity of most users, and the fact that typically only one user can consume an advertised item. In this work, we address these challenges by choosing models and evaluation procedures that are considered accurate, diverse, and efficient (in terms of memory and time consumption during training and prediction). This paper aims to benchmark various recommendation methods for job classifieds, using OLX Jobs as an example, to enhance the conversion rate of advertisements and user satisfaction. In our research, we implement scalable methods and represent different approaches to the recommendations: Alternating Least Square (ALS), LightFM, Prod2Vec, RP3Beta, and Sparse Linear Methods (SLIM). We conducted A/B tests by sending millions of messages with recommendations to perform online evaluations of selected methods. In addition, we have published the dataset created for our research. To the best of our knowledge, this is the first dataset of its kind. It contains 65,502,201 events performed on OLX Jobs by 3,295,942 users who interacted with (displayed, replied to, or bookmarked) 185,395 job ads over two weeks in 2020. We demonstrate that RP3Beta, SLIM, and ALS perform significantly better than Prod2Vec and LightFM when tested in a laboratory setting. Online A/B tests also show that sending messages with recommendations generated by the ALS and RP3Beta models increases the number of users contacting advertisers. Additionally, RP3Beta had a 20% more significant impact on this metric than ALS.

Keywords: job recommendations; classifieds; collaborative filtering; scalability testing; A/B tests; recommender systems; matrix factorization; graph-based recommender systems; OLX job interactions dataset; diversity metrics



Citation: Kwieciński, R.; Górecki, T.; Filipowska, A.; Dubrov, V. Job Recommendations: Benchmarking of Collaborative Filtering Methods for Classifieds. *Electronics* **2024**, *13*, 3049. <https://doi.org/10.3390/electronics13153049>

Academic Editors: Marco Lombardi, Domenico Santaniello, Zilu Liang and Muhammad Khan

Received: 30 May 2024

Revised: 24 July 2024

Accepted: 27 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Online classifieds are websites where advertisers post advertisements concerning the sale or rental of services or products [1]. This paper addresses online advertising using the example of OLX Jobs.

Classifieds strive to address the needs of both advertisers and responders to online ads. To match these groups, better recommendation methods are becoming more critical. Compared to traditional e-commerce (e.g., Amazon (<https://www.amazon.com/>, accessed on 26 June 2024) or eBay (<https://www.ebay.com/>, accessed on 26 June 2024)), classifieds must face different challenges, which we address in our evaluation procedure. One is the temporal visibility of ads, which we decided to address by frequently retraining the model. Hence, the model's efficiency, which we report, is essential to the evaluation. Another challenge is that an advertiser usually needs only one user to complete the transaction. This aspect was addressed by including diversity analysis in our assessment.

In our research, we focus on approaches emerging from collaborative filtering. Collaborative filtering models have proven their accuracy, even though they do not utilize additional information about users or items, which is sometimes hard to obtain or process in practice. This paper aims to evaluate different recommendation methods for use in the classifieds sector. The high sparsity of our dataset, along with the significant number of ads and users, contributes to the complexity of the research problem.

Below, we list our main contributions.

- We comprehensively evaluate diverse, scalable, and collaborative filtering approaches in a laboratory setting regarding accuracy, diversity, and efficiency. We evaluate the accuracy for different groups of users, depending on the number of items they interacted with, together with statistical analysis of differences. The proposed evaluation methodology addresses the major challenges of classifieds.
- We present the results of two online A/B tests demonstrating the impact of the ALS and RP3Beta models on OLX users. More than 1 million users participated in each test. The reported impact may be used to estimate the importance of recommender systems in the classifieds domain and enable other companies to make more informed decisions about developing such models.
- We publish and present a job interactions dataset. To our knowledge, this is the largest publicly available dataset on job interactions.

This paper consists of seven sections. Section 2 is a literature review on underlying developments. Section 3 describes the dataset developed and made publicly available on Kaggle. The design of the lab experiment, as well as the implementation of methods, are described in Section 4. Section 5 focuses on the results of the lab evaluation of our implementations as well as the statistical importance of the results. Section 6 presents the results of the online implementations performed during A/B tests with users, as well as a discussion of the results. Conclusions are presented in Section 7.

2. Related Work

In this section, we begin by outlining the unique characteristics of classifieds and job domains. We then provide an overview of recommender systems. Next, we discuss the factors that influence our selection of models and provide a detailed description of each model. Finally, we specify the research gaps.

2.1. Classified ad Sites

Classified ad sites (classifieds) are online versions of newspaper advertisement sections, with ads concerning vehicles, real estate, employment, pets, etc. Classifieds are a particular type of e-commerce site, as they do not support the finalization of transactions between parties—they are used instead to communicate about the product or service and to set up meetings while the transaction is carried out offline; this distinguishes them from traditional e-commerce sites such as eBay or Amazon [2].

Our research is focused on OLX, which has a presence in over 30 countries worldwide. One of the ad types published on classifieds is job offers. In our research, we identified several characteristics that distinguish job ads on classifieds from, for example, car or real estate ads:

- The requirements in an ad indicate the type of user who may respond to the ad—not all users possess the required competencies.
- More intensive involvement of the user during the search process, as finding a job is a primary need compared to, for example, buying a car.
- The job location dramatically impacts the user—to buy a mobile phone or a car, a user may travel or use a parcel service to deliver the item. Still, a job requires relocation if it is distant from the user's home.
- From a technical perspective, a job description has many qualitative aspects, and depending on the position, the differences may be significant even between ads from the same company.

On top of these features, during our research, we also identified the following challenges for recommendation methods on classifieds that influenced our implementation of the methods:

- The number of users and offers: with OLX Jobs, we dealt with millions of users and tens of thousands of offers, both constantly changing.
- A user does not have to create a profile to interact with ads, meaning that information from a profile cannot be used for recommendation, and recommendations are determined by user behavior during viewing.
- An ad usually has a limited number of visitors (interacting with the ad), as an ad usually concerns a unique offer. After the need is addressed, the ad is disabled.
- There is no information on whether a transaction occurred; on classifieds, conversion relates to receiving an answer to a published ad, as any transaction is offline.

These challenges create difficulties not only for implementation but also for testing the accuracy of the proposed methods. We considered these challenges while preparing the dataset and implementing and evaluating the methods described in this paper.

2.2. Recommendation Systems

It has been shown that the problem of overchoice [3] can decrease the consumption of products [4] and reduce revenue to the site [5]. Providers can ask their users for additional information about their expectations regarding items to reduce the number of choices presented. Even if the user is willing to spend time providing this information, the number of possibilities is usually too large to present to the user. To address this problem, many companies, such as Netflix [6] and Amazon [7], have created personalized recommendation systems whose role is to suggest relevant items to users [8].

Recommendations are used in multiple domains, including music, tourism, fashion, and food. Numerous studies have been made within the domain of e-commerce [9], classifieds [10], and job recommendations [11–13]. Li et al. [14] significantly improved their recommendations on LinkedIn by utilizing deep transfer learning to create domain-specific job-understanding models. Zhu et al. [15] proposed a cross-domain recommendation model utilizing a heterogeneous graph and community detection algorithm to recommend courses and jobs. Lacic et al. [16] evaluated different autoencoder architectures for session-based job recommendations and presented the results on popular job recommendation datasets.

Typically, we distinguish two approaches for providing recommendations: content-based recommendations and collaborative filtering [8].

In content-based recommendations [17], we utilize information about users (e.g., hobby, education, skills) and items (e.g., price, location, category). These systems usually use the interaction history (ratings, visits, purchases, and replies) of a single user but do not consider the ratings of all users simultaneously.

Collecting/processing information about users and items is not always feasible. In the collaborative filtering approach [18,19], we only utilize information about user interactions between items. Even without information on user features and item features, these systems can provide very accurate recommendations. We chose this approach because of the lack of user features and the difficulty of outperforming collaborative filtering by utilizing item features in a production environment (we elaborate on this in Section 6.3).

One of the most significant shortcomings of collaborative filtering is the impossibility of providing recommendations regarding users and items without interactions, known as the cold-start problem [20]. One possible solution to reducing this problem is the frequent retraining of the model. With many user interactions, this strategy requires a focus on the scalability of the model [21].

Another important factor determining the model choice is the type of interaction between a user and an item [20]. We call the feedback explicit when the user explicitly indicates their preference toward the item (e.g., rates it on a scale of 1–5). Another type is implicit feedback, when we can only assume the user's preference toward the item based

on user behavior (e.g., buying an item or watching a movie). Implicit feedback is less accurate, more abundant, and usually expresses only favorable preferences [22].

Laboratory evaluations of recommender systems are usually based on accuracy metrics [23]. Recently, much attention has been placed on other aspects, such as diversity, uncertainty, novelty, and coverage [24,25]. With classifieds, an item can usually be consumed by only one user, so we should avoid recommending the same item to a significant number of users. Hence, focusing on catalog coverage [26] is essential. One of the most challenging tasks in evaluating recommender systems is finding the offline metric most correlated with the online goal [27].

In this paper, we benchmark collaborative filtering models for a sizable implicit feedback dataset. We compare the accuracy, diversity, and efficiency of selected recommendation methods in a laboratory setting and then prove the methods' quality in online tests with users.

2.3. Recommendation Methods

Below, we list the most important aspects impacting method selection for our research.

- **Popularity in the literature.** We selected methods whose performances were thoroughly evaluated. All but one of the selected methods have been cited at least hundreds of times.
- **Popularity in industry.** Evaluating this aspect was challenging because companies rarely share the results of conducted experiments. Some popularity indicators include implementation availability and the number of people who use it (e.g., reflected in the number of stars and forks on GitHub). We also considered the experience of data scientists working within our company, who deployed most of the selected methods for some other use cases in the past.
- **Scalability.** We only considered solutions capable of handling millions of users and items. Additionally, we reduced the cold-start problem by frequently retraining the model. Hence, we had to choose techniques to be trained on our dataset within a few hours.
- **Diversity of approaches.** We wanted to test methods representing different "families" to assess their applicability to classifieds. We planned to identify the most promising family of approaches to continue research in this direction.

Finally, ALS, LightFM, Prod2Vec, RP3Beta, and SLIM were selected for testing and evaluation. The following subsections describe the reasons for choosing each specific method and the method definition. A common limitation of these methods is their inability to recommend items that are not visited by any user and generate recommendations for users who have not visited any item (cold-start problem, [21]). With our dataset, without user and item features, it is impossible to overcome this problem by any personalized method.

Selected methods have already been compared across different domains. Anelli et al. [28] compared RP3Beta, SLIM, ALS, and BPRMF (which is equivalent to LightFM when using the BPR loss function) on three datasets: Amazon Digital Music, Epinions, and MovieLens 1M, in terms of accuracy, diversity, and efficiency. The numbers of interactions in these datasets are, respectively, 145,523, 300,475, and 571,531. Our study utilized a dataset containing tens of millions of interactions, similar to those in OLX production systems. Additionally, we provide the results of an online evaluation.

Dacrema et al. [29] compared the accuracy of SLIM, RP3Beta, and ALS on several datasets (CiteULike-a, MovieLens 1M, Pinterest, Amazon Music, Amazon Movies, MovieLens 20M, Netflix, Yelp, and Gowalla). They showed that each of these algorithms outperformed the other two on at least one dataset. Some of their findings were further confirmed on MovieLens 1M and Pinterest datasets by Anelli et al. [30]. Zhu et al. [31] compared BPRMF, SLIM, and Prod2Vec on three datasets (Amazon Books, Yelp, and Gowalla), showing the superiority of SLIM. Dong et al. [32] analyzed the accuracy of RP3Beta, SLIM, and ALS on nine datasets, with SLIM providing the best results on eight datasets and ALS

on one. Unlike our study, these papers neither compared the selected methods in terms of diversity or efficiency nor conducted an online evaluation.

Recently, several recommendation methods based on neural networks have been proposed. It has been shown that adequately determined and tuned simple models often outperform them [28,31–33]. Additionally, due to computational complexity, models based on neural networks are less frequently tested on datasets containing tens of millions of interactions [28]. Hence, we leave it for future research to propose more advanced and scalable methods that provide superior results on our dataset.

2.3.1. Matrix Factorization Models

Matrix factorization techniques have been successfully utilized in recommender systems since the Netflix Prize competition [34]. The main idea is to approximate the sparse user–item interaction matrix as a product of two smaller, dense matrices. We evaluate two matrix factorization approaches: LightFM and ALS.

LightFM [35] model was proposed to overcome the cold-start problem of matrix factorization methods. In our case, it reduces to classical matrix factorization without additional information about users and items. Despite this, we decided to use this approach because of its efficient official implementation in supporting multiple loss functions: logistics, BPR [36], WARP [37], and k -OS WARP [38].

Let $\mathcal{U}$ be a set of users and $\mathcal{I}$ a set of items. For a given user $u \in \mathcal{U}$ and item $i \in \mathcal{I}$, we define $r_{ui} = 1$ if user u interacts with item i , and $r_{ui} = 0$, otherwise. The LightFM model predicts the score r_{ui} as follows:

$$\hat{r}_{ui} = \mathbf{x}_u \cdot \mathbf{y}_i + b_u + b_i,$$

where $\mathbf{x}_u$ and $\mathbf{y}_i$ denote user u 's and item i 's k -dimensional latent representations, and b_u and b_i are user u 's and item i 's biases. All of these parameters are learned during the training.

The logistic loss function should be used when both positive and negative feedback types are available, so it is inappropriate for our dataset, which we confirmed experimentally during the hyperparameter tuning.

The BPR, WARP, and k -OS WARP loss functions are pairwise learning-to-rank approaches that are more relevant for our top- k recommendations task. They try to modify model parameters to maximize the difference between the scores of the items $i \in \mathcal{D}_u$ with which user u interacts and of the items $i \notin \mathcal{D}_u$ with which the user does not interact with.

Specifically, in the BPR loss function [36], we attempt to maximize the expression, as follows:

$$\sum_{(u,i,j) \in \mathcal{S}} \ln \sigma(\hat{r}_{ui} - \hat{r}_{uj}) - \lambda_{\Theta} \|\Theta\|^2,$$

where

$$\mathcal{S} = \{(u, i, j) | u \in \mathcal{U}, i \in \mathcal{D}_u \text{ and } j \notin \mathcal{D}_u\},$$

$$\sigma(x) = \frac{1}{1 + e^{-x}},$$

λ_{Θ} is a model-specific regularization parameter, Θ is a parameter vector, and $\|\cdot\|$ is the Euclidean norm.

We present WARP [37] as a case of k -OS WARP [38]. The k -OS WARP loss function is a sum of losses for each user:

$$\sum_{u \in \mathcal{U}} L(\hat{\mathbf{r}}_u, \mathcal{D}_u),$$

where $\hat{\mathbf{r}}_u$ is a vector of estimated scores $\hat{r}_{ui}$ for each item $i \in \mathcal{I}$.

For a given user u , we order the interacted items $\mathcal{D}_u$ in decreasing order of the estimated scores. Namely, we order the items as $i_1, i_2, \dots, i_{|\mathcal{D}_u|}$, where the indices are chosen in such a way that we have the following:

$$\hat{r}_{ui_1} \geq \hat{r}_{ui_2} \geq \dots \geq \hat{r}_{ui_{|\mathcal{D}_u|}}.$$

Then, we have the following:

$$L(\hat{\mathbf{r}}_u, \mathcal{D}_u) = \frac{1}{Z} \sum_{s=1}^{|\mathcal{D}_u|} P(s) \Phi(\text{rank}_{i_s}(\hat{\mathbf{r}}_u)),$$

where $\text{rank}_i(\hat{\mathbf{r}}_u) = \sum_{j \in I \setminus \mathcal{D}_u} \mathbb{1}(1 + \hat{r}_{uj} \geq \hat{r}_{ui})$, $\mathbb{1}$ is an indicator function, $\Phi(n) = \sum_{m=1}^n \frac{1}{m}$, and $Z = \sum_{s=1}^{|\mathcal{D}_u|} P(s)$ normalizes the weights induced by P .

In practice, we sample a positive item i concerning the weighting function P and randomly pick unseen items j until, after N samplings, $1 + \hat{r}_{uj} \geq \hat{r}_{ui}$. Then, we perform a gradient step to minimize $\Phi(\lfloor \frac{|I \setminus \mathcal{D}_u|}{N} \rfloor)(1 - \hat{r}_{ui} + \hat{r}_{uj})$.

Finally, the k -OS WARP loss is achieved for $P(k) = 1$ and $P(m) = 0$ if $m \neq k$. The WARP loss is achieved when we set $P(m) = 1$ for $m \in \mathbb{N}$. We can skip ordering positive items from $\mathcal{D}_u$ to simplify the computations.

In our research, all loss functions (logistic, BPR, WARP, and k -OS WARP) were treated as hyperparameters of the LightFM model and optimized during training.

The ALS model (alternating least squares), also known as WRMF—weighted regularized matrix factorization [39], is an example of a matrix factorization method designed for implicit feedback datasets. We chose this method because of its proven performance, scalability, and popularity in research and industry. Additionally, the ALS model has already been implemented and used in OLX.

We define the confidence of an observation as $c_{ui} = 1 + \alpha r_{ui}$, where α is a hyperparameter. We also define the preference as $p_{ui} = \mathbb{1}(r_{ui} > 0)$. Then, in ALS, we optimize the following expression:

$$\min_{\mathbf{x}_*, \mathbf{y}_*} \sum_{u,i} c_{ui} (p_{ui} - \mathbf{x}_u^\top \mathbf{y}_i)^2 + \lambda (\sum_u \|\mathbf{x}_u\|^2 + \sum_i \|\mathbf{y}_i\|^2),$$

where $\mathbf{x}_u$ is user u 's f -dimensional embedding, $\mathbf{y}_i$ is item i 's f -dimensional embedding, and λ is a regularization parameter. The model's name refers to the optimized method for this function. The alternating least squares procedure assumes that user or item latent factors are fixed and analytically determine the others. The procedure is efficiently repeated a few times [39,40].

In matrix factorization approaches, the embedding for each user is learned separately, which might result in high time and memory utilization if the number of users is vast. Additionally, the recommendations might lack explainability because two users with the same interactions might receive different recommendations. Other considered models (SLIM, RP3Beta, Prod2Vec) provide the same set of recommendations for users with the same interactions.

2.3.2. Neighborhood-Based Models

The SLIM model (sparse linear methods for Top-N recommender systems [41]) is based on item-based nearest neighbors regression [8]. We chose this method because of its advantages over item-based k -nearest-neighbors and matrix factorization models [41] and good performance against non-neural and neural approaches [29].

In this model, we approximate the user–item matrix $\mathbf{R}$ by learning the item–item sparse similarity matrix $\mathbf{W}$. Explicitly, $\mathbf{W}$ is learned by minimizing the following expression:

$$\frac{1}{2} \|\mathbf{R} - \mathbf{R}\mathbf{W}\|_F^2 + \frac{\beta}{2} \|\mathbf{W}\|_F^2 + \lambda \|\mathbf{W}\|_1,$$

where $w_{ij} \geq 0$ and $w_{ii} = 0$ for all $i, j, 1 \leq i, j \leq |I|$, $\|\cdot\|_F$ is the matrix Frobenius norm:

$$\|\mathbf{W}\|_F = \sqrt{\sum_{i=1}^{|I|} \sum_{j=1}^{|I|} w_{ij}^2},$$

$\|\cdot\|_1$ is the entry-wise ℓ_1 -norm:

$$\|\mathbf{W}\|_1 = \sum_{i=1}^{|I|} \sum_{j=1}^{|I|} |w_{ij}|$$

and β and λ are regularization parameters.

We can see that each column of $\mathbf{W}$ can be learned independently, which improves the scalability of this model [41].

A limitation of this method is that it does not generate user and item embeddings, which could be helpful for other tasks (e.g., clustering of users or items). We can treat the columns of the matrix $\mathbf{W}$ as item embeddings, but they are sparse $|I|$ -dimensional vectors, so they probably cannot be directly utilized in other tasks.

2.3.3. Graph-Based Approaches

RP3Beta [42] is a graph-based collaborative filtering approach. Similarly to Dacrema et al. [29], we present this model as an extension of the P3alpha model [43]. We consider a bipartite and undirected graph $G = (\mathcal{V}, \mathcal{E})$, where each vertex $v \in \mathcal{V}$ represents either a user or item and each edge $\{v_u, v_i\} \in \mathcal{E}$ represents the interaction between a given user u and a given item i . Let $\mathbf{A}$ be the adjacency matrix and $\mathbf{D}$ the degree matrix of the graph G . Then, the scores of P3alpha are stored in the matrix

$$((\mathbf{D}^{-1}\mathbf{A})^{\circ\alpha})^3,$$

where $\circ$ denotes the element-wise power of a matrix (Hadamard power). More precisely, the score $\hat{r}_{ui}$ is stored on the intersection of the row representing user u and the column representing item i . We can see that this score is a sum of scores assigned to paths of length 3 connecting user u and item i in the graph G .

In RP3Beta, the score is computed with P3alpha and divided by the popularity of the items raised to the power of β to mitigate the popularity bias.

We chose this model because of its simplicity, scalability, and proven performance [29,30], especially given the high sparsity of our dataset, which significantly decreases memory utilization when training the model. This enables us to directly compute and store the item-item similarity matrix in memory instead of the random walks approximation proposed in the original paper.

RP3Beta, similar to SLIM, does not generate user and item embeddings. Another related limitation is its inability to recommend items whose distance on the user-item bipartite graph is greater than three from a given user. Specifically, item i can be recommended to user u only if at least one user u' has visited item i and at least one of the items visited by user u . Additionally, the RP3Beta model does not have parameters optimized during training; hence, it might be unable to exploit the complex relations in data.

2.3.4. Prod2Vec

The Prod2Vec [44] (or Item2vec [45]) model is based on the Word2Vec model [46], widely used in natural language processing. The model focuses on learning vector representations of items using a sequence of items as a “sentence” and items within that sequence as “words”. This method was chosen because it is scalable and very different from the compared methods. Additionally, this approach is based on the current state-of-the-art hybrid deep neural network recommender system used in other categories at OLX.

Prod2Vec uses the skip-gram model by maximizing the objective function over the set S of item sequences, defined as follows:

$$L = \sum_{s \in S} \sum_{i_k \in s} \sum_{\substack{-c \leq l \leq c \\ l \neq 0}} \log \mathbb{P}(i_{k+l}|i_k),$$

where c is the context length for item sequences, and items from the same sequence are ordered arbitrarily. The probability $\mathbb{P}(i_{k+l}|i_k)$ of observing a neighboring item i_{k+l} given the current item i_k is defined using the soft-max function:

$$\mathbb{P}(i_{k+l}|i_k) = \frac{\exp(\mathbf{y}_{i_k}^\top \mathbf{y}'_{i_{k+l}})}{\sum_{i=1}^{|I|} \exp(\mathbf{y}_{i_k}^\top \mathbf{y}'_i)},$$

where $\mathbf{y}_i$ and $\mathbf{y}'_i$ are the input and output vector representations of item i , and $|I|$ is the number of unique items in the vocabulary. Prod2Vec models the context of an item sequence, where items with similar contexts (i.e., with similar neighboring interactions) have similar vector representations.

Before generating recommendations, we normalize all input vector representations, which improves efficiency. To generate recommendations, we represent the user as an average of input vector representations of interacted items. Then, we search for items whose input representations have the highest cosine similarity with the user representation (k -nearest neighbors in the latent space).

A potential limitation of this method is a strict dependency between user and item representations. Some interactions should potentially have a greater impact on user representations than others (which is possible in matrix factorization models).

2.4. Research Gap

This paper aims to benchmark different recommendation methods in the job classifieds domain, addressing a specific research gap. To the best of our knowledge, the recommendation methods addressed have never been tested or benchmarked on a large scale while working online in the job domain. In this work, we compare these methods in a laboratory setting using the dataset we published and in online A/B tests carried out in a production environment.

3. OLX Job Interactions Dataset

This research provides a commercial dataset that contains 65,502,201 events performed on OLX Jobs by 3,295,942 users who interacted with 185,395 job ads in two weeks in 2020. We published the dataset on Kaggle (<https://www.kaggle.com/olxdatascience/olx-jobs-interactions>, accessed on 26 June 2024).

Each row of the dataset represents the event a given user performs regarding a given ad at a given time (Table 1). The following events involving user behaviors were included:

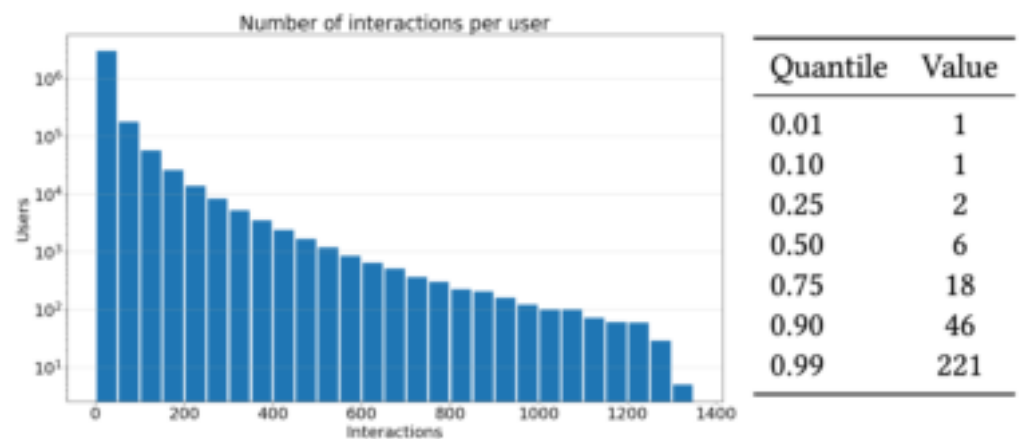
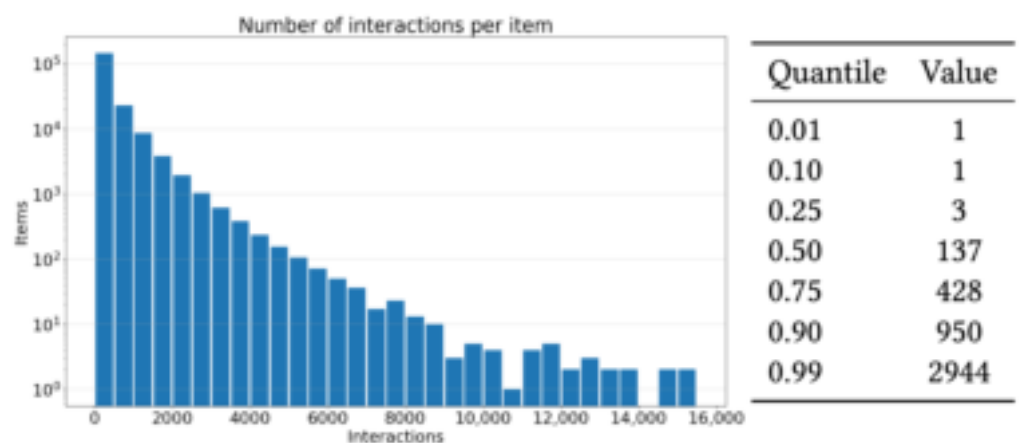
- click: the user visited the item detail page;
- bookmark: the user added the item to bookmarks;
- chat_click: the user opened the chat to contact the item's owner;
- contact_phone_click_1: the user revealed the phone number attached to the item;
- contact_phone_click_2: the user clicked to make a phone call to the item's owner;
- contact_phone_click_3: the user clicked to send an SMS message to the item's owner;
- contact_partner_click: the user clicked to access the item's owner's external page;
- contact_chat: the user sent the item's owner a message.

Table 1. Example rows from the dataset.

User	Item	Event	Timestamp
1745587	168661	click	1582216025
843008	62838	click	1582485868
14285	30469	bookmark	1582247367
1142944	80122	click	1581805847
2835659	23728	chat_click	1582397836

Source: own study.

The data sparsity (% of missing entries in the user–item matrix) equals 99.9923%, which is very low but typical for classifieds (for comparison, the data sparsity on the MovieLens 20M Dataset is 99.47%, making it almost 70 times denser). Distributions of interactions per user, interactions per item, and events are provided in Figures 1 and 2, and Table 2, respectively. We used a logarithmic scale because the distributions are strongly right-skewed.

**Figure 1.** Number of interactions per user. Source: own study.**Figure 2.** Number of interactions per item. Source: own study.

According to De Ruijt and Bhulai [47], the most popular datasets used for the evaluation of job recommendations systems are CareerBuilder12 (<https://www.kaggle.com/competitions/job-recommendation>, accessed on 16 June 2024), RecSys16 [12], and RecSys17 [13]. Table 3 shows the statistics of considered datasets. The number of unique interactions is the number of unique user–item pairs where the user interacted with the item. Even though researchers use the RecSys16 and the RecSys17 datasets, they were published during data science competitions under strictly limited conditions and are no longer available. Hence, we could not calculate the number of unique interactions for these

datasets. We did observe that the OLX job interactions dataset has the highest number of interactions and types among the considered datasets.

Table 2. Distribution of events.

Action	Frequency (%)
click	89.794
contact_phone_click_1	2.628
bookmark	2.511
chat_click	2.136
contact_chat	1.448
contact_partner_click	0.701
contact_phone_click_2	0.679
contact_phone_click_3	0.103

Source: own study.

Table 3. Comparison between job recommendation datasets. The letter *D* denotes the standard deviation.

	OLX Jobs Interactions	CareerBuilder12	RecSys16	RecSys17
Interactions	65.50 M	1.60 M	8.83 M	8.27 M
Unique interactions	47.17 M	1.60 M	-	-
Users	3.30 M	0.3 2M	1.37 M	1.50 M
Average number of unique interactions per user	14.31 (<i>D</i> = 29.23)	4.99 (<i>D</i> = 11.42)	<6.46	<5.51
Items	0.19 M	0.37 M	1.36 M	1.31 M
Average number of unique interactions per item	254.42 (<i>D</i> = 426.02)	4.38 (<i>D</i> = 8.19)	<6.50	<6.31
Density of the interaction matrix (%)	0.0077%	0.0014%	<0.0005%	<0.0004%
Types of interactions	8	1	4	5
Timestamps	✓	✓	✓	✓
User features	✗	✓	✓	✓
Item features	✗	✓	✓	✓

Source: own study.

It should be noted that maintaining the confidentiality of ads and users was a priority when preparing this dataset. The measures taken to protect privacy included the following:

- Original user and item identifiers were replaced with unique random integers;
- Some undisclosed constant integer was added to each timestamp;
- Some fractions of interactions were filtered out;
- Some additional artificial interactions were added.

4. Experimental Setup

4.1. Implementation

We implemented the methods mentioned above based on previous developments. Details on how our implementations differ from the previously tested approaches are provided in Table 4. The implementation of the methods for reproducibility of the results is available on GitHub (<https://github.com/rob-kwiec/olx-jobs-recommendations>, accessed on 29 May 2024).

Table 4. Methods re-implemented for the research.

Method	Family of Methods	Source	Difference from the Original Method
LightFM	Matrix factorization	Implementation based on: [48]. Supporting paper: [35].	None
ALS	Matrix factorization	Implementation based on: [49]. Supporting papers: [39,40].	None
SLIM	Neighborhood-based	Sources that inspired our implementation: [50,51]. Supporting paper: [41].	None
RP3Beta	Graph-based	Source that inspired our implementation: [52]. Supporting papers: [29,53].	Performed direct computations on sparse matrices instead of random walks approximation.
Prod2Vec	Word2Vec	Implementation based on: [54]. Supporting papers: [44,45].	Sequences of interactions, ordered by a timestamp. Representing the user as an average of the representations of interacted items. We also experimented with CBOW (continuous bag of words).

Source: own study.

4.2. Laboratory Research and Online Testing Goals

Our ultimate business goal was to increase the number of users applying for jobs through OLX Jobs. To achieve that, we try to solve a ranking problem; more precisely, we recommend ten items with the highest chance of receiving a given user's interaction. The issue of finding an offline metric most correlated with the application rate in the job domain was considered by Mogenet et al. [27]. Since our goal is similar, we followed their results by optimizing for precision@10 (precision for the first ten recommended items [23]). However, we also report and discuss other accuracy, diversity, and efficiency metrics.

We are focused on users actively looking for jobs by interacting with job ads. Therefore, we do not consider users without interactions (cold-start users).

4.3. Data Split

The dataset described in Section 3 was split into training-validation and test datasets as required by the research methodology. The test set includes 20% of the newest interactions. This means that out of 14 days, approximately 3 days were included in the test set (see Figure 3). This way of splitting the dataset into training and test sets is the most realistic setting according to Meng et al. [55], who studied several splitting strategies for evaluating recommendation systems.

In both datasets, we also limited the number of interactions to unique interactions between a user and an item. If a user interacted more than once with an item, only the first interaction was counted, and the timestamp of that interaction was associated with the item. We also did not distinguish between types of interactions. In addition, two more modifications were applied: only users who appeared in the training and validation dataset were included in the test set (because we could not provide personalized recommendations to other users without additional knowledge about them), and we filtered out all user-item pairs that were present in the training set (to avoid recommending items that had already been seen). As a result, we ended up with about 38 M rows in the training and validation set and about 6 M rows in the test set. More statistics of these two datasets are presented in Table 5.

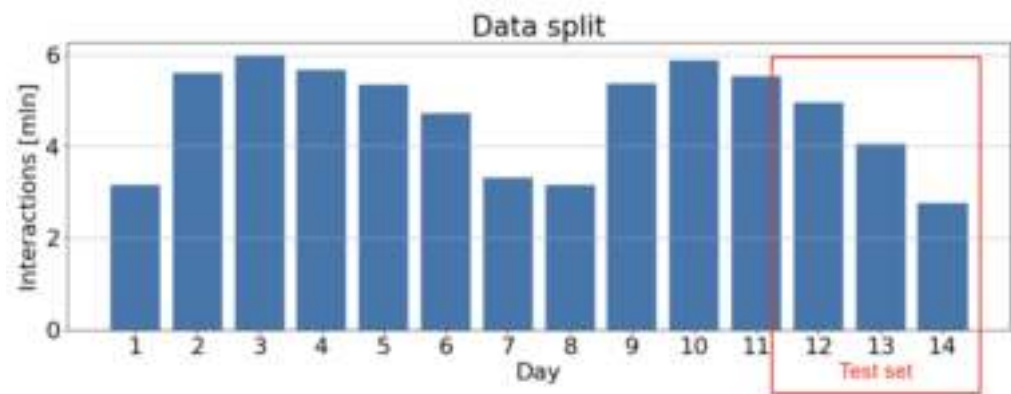


Figure 3. Dataset preparation (split into training–validation and testing, depending on the number of days). Source: own study.

Table 5. Comparison between training/validation and test sets. The letter *D* denotes standard deviation.

	OLX Job Interactions	Training and Validation Set	Test Set
Interactions	65.50 M	52.40 M	6.11 M
Unique interactions	47.17 M	38.25 M	6.11 M
Users	3.30 M	2.83 M	0.62 M
Average number of unique interactions per user	14.31 (<i>D</i> = 29.23)	13.50 (<i>D</i> = 26.11)	9.87 (<i>D</i> = 14.38)
Items	0.19 M	0.18 M	0.13 M
Average number of unique interactions per item	254.42 (<i>D</i> = 426.02)	214.05 (<i>D</i> = 365.93)	47.51 (<i>D</i> = 83.84)
Density of the interaction matrix (%)	0.0077%	0.0076%	0.0077%

Source: own study.

4.4. Model Tuning

For the sake of cost and time, when performing the optimization of parameters, we made a restriction to 20% of users and 20% of items of the training and validation set. Then, we split the restricted dataset into the training–validation dataset with the rules described in the section above. For each model, there were 100 iterations chosen by Bayesian optimization using Gaussian processes implemented in scikit-learn (https://scikit-optimize.github.io/stable/modules/generated/skopt.gp_minimize.html, accessed on 29 May 2024). The configuration file defines spaces for hyperparameters and is available in our code repository (<https://github.com/rob-kwiec/olx-jobs-recommendations/blob/main/src/tuning/config.py>, accessed on 29 May 2024). For example, for RP3Beta, we restricted ourselves to searching for α and β in the interval $[0, 2]$.

Recommendations in the optimization process were prepared for 30,000 users, and as mentioned above, the optimization score used was precision@10. Figure 4 shows the precision score depending on the parameters. It is visible that ALS was not highly optimized within the process, but for other methods, this optimization was crucial to improving precision. Hyperparameters for all models are presented in Table 6.

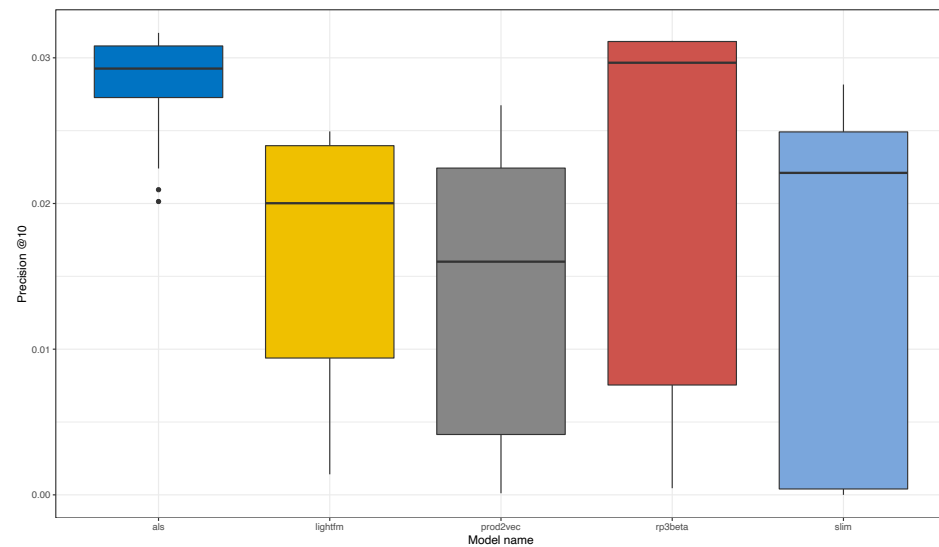


Figure 4. Precision for each model, depending on the parameters. Source: own study.

Table 6. Model hyperparameters.

Model	Model Hyperparameters
als	{‘factors’: 357, ‘regularization’: 0.001, ‘iterations’: 20, ‘event_weights_multiplier’: 63}
lightfm	{‘no_components’: 512, ‘k’: 3, ‘n’: 20, ‘learning_schedule’: ‘adadelta’, ‘loss’: ‘warp’, ‘max_sampled’: 61, ‘epochs’: 11}
prod2vec	{‘vector_size’: 168, ‘alpha’: 0.028728, ‘window’: 20, ‘min_count’: 16, ‘sample’: 0.002690026, ‘min_alpha’: 0.0, ‘sg’: 1, ‘hs’: 1, ‘negative’: 200, ‘ns_exponent’: −0.16447846705441527, ‘cbow_mean’: 0, ‘epochs’: 22}
RP3Beta	{‘alpha’: 0.61447198, ‘beta’: 0.1443548}
slim	{‘alpha’: 0.00181289, ‘l1_ratio’: 0.0, ‘iterations’: 3}

Source: own study.

4.5. Model Evaluation

We evaluate our models in terms of accuracy, diversity, and efficiency.

The accuracy of the recommendation method on implicit feedback datasets is typically evaluated by examining which items from the list of the top k recommendations the user interacted with in the test set [8]. Many evaluation metrics determine the metrics for each user separately and present the mean as the final evaluation score of the method [23]. The recommendation system may be considered a classifier, where the user’s interactions in the test set are positive examples, and the top k recommendations are items classified as positive by the model. Therefore, methods used for evaluating classifiers can be directly applied to recommender systems. For instance, the value of $\text{precision}@k$ for a given user is the number of recommended items the user interacted with in the test set divided by the number k of items recommended to that user. Other metrics discussed in this work were described by Tamm et al. [23].

In the classifieds domain, primarily the jobs domain, only one user is usually needed to complete a transaction regarding a given ad. In this case, the diversity of recommendations is even more critical than in other domains (like music, movies, or other e-commerce, excluding classifieds). To assess this aspect, we use three measures: test coverage, Shannon entropy [56], and the Gini index [56]. Test coverage refers to the fraction of test items recommended to at least one user. With Shannon entropy and the Gini Index, we ignore items outside the test set. Greater test coverage and Shannon entropy values indicate higher diversity, whereas greater values of the Gini index indicate lower diversity.

The final laboratory evaluation step focuses on efficiency. For this purpose, we utilized an AWS SageMaker ml.m5.4xlarge instance equipped with 64 GB of RAM and 16 vCPUs, featuring an Intel(R) Xeon(R) Platinum 8259CL CPU @ 2.50GHz, running on Amazon Linux AMI 2018.03. All methods were evaluated concerning three functions: data preprocessing, model fitting, and item recommendation. The execution time was calculated for the total time needed to compute the model (in hours). The memory peak was calculated in GB, and maximum memory usage was considered when executing a given function. This evaluation was performed using the tracemalloc library (<https://docs.python.org/3/library/tracemalloc.html>, accessed on 29 May 2024).

5. Laboratory Evaluation

This section describes the laboratory evaluation of the methods discussed in the developed dataset. We used the best-performing hyperparameters found in the previous section. Each model was trained on the entire training and validation dataset, and recommendations were produced for all 619,389 users from the test set. To better present the values of each offline metric, we compared the described methods with simple, non-personalized baselines: the most popular product approach and random recommendations.

5.1. Accuracy

All metrics in this section consider the first ten recommendations, e.g., precision@10, recall@10. Higher metric values indicate a higher quality of recommendations. The values should not be directly compared with the results achieved on other datasets because metrics heavily depend on the distribution of the dataset (for example, high sparsity) and the train/test splitting strategy [55].

5.1.1. Evaluation of Accuracy

Table 7 presents the outcomes of the evaluation. Most of the observed differences are statistically significant due to the large number of evaluated users. We elaborate more on the statistical significance of the results in Section 5.1.2.

Table 7. Accuracy: evaluation results. All presented metrics were described by Tamm et al. [23]. For the mAP metric, we used the variant with $x = \min(k, r)$.

Metric	RP3Beta	SLIM	ALS	Prod2Vec	LightFM	Most Popular	Random
precision	0.0484	0.0472	0.0434	0.0368	0.0359	0.0012	0.00006
recall	0.0783	0.0736	0.0657	0.0580	0.0564	0.0012	0.00005
ndcg	0.0759	0.0721	0.0657	0.0567	0.0545	0.0016	0.00007
mAP	0.0393	0.0365	0.0329	0.0282	0.0264	0.0006	0.00002
MRR	0.1365	0.1314	0.1230	0.1065	0.1034	0.0038	0.00019
LAUC	0.5391	0.5368	0.5328	0.5289	0.5281	0.5006	0.49999
HR	0.3131	0.3066	0.2878	0.2537	0.2547	0.0112	0.00059

Source: own study.

Our approaches significantly outperform random and most popular item recommendations. The low performance of the most popular approach results from the nature of classifieds, primarily the domain of the job, where users are mostly interested in a specific location and category. Building more sophisticated, personalized recommendation systems is reasonable in such a scenario.

It is demonstrated that RP3Beta outperforms other approaches in terms of accuracy metrics. We believe that this is related to the high sparsity of our dataset. RP3Beta calculates the recommendations deterministically (by leveraging paths of length three on a user-item bipartite graph). All other approaches utilize machine learning techniques to find user representations (ALS, LightFM), item representations (ALS, LightFM, Prod2Vec), or direct item-item similarities (SLIM), which might be in cases where users or items have a low number of interactions.

5.1.2. Statistical Comparison for Precision@10

We present a detailed statistical comparison for our main metric, precision@10, to identify differences between the methods. To begin with, we test the null hypothesis that all methods perform the same and the observed differences are merely random (omnibus test). The Friedman test [57,58] with the Iman and Davenport [59] extension is probably the most popular omnibus test, and it is usually a good choice when comparing more than five different algorithms [60,61]. Let R_{ij} be the rank of the j th of K methods on the i th of N samples and

$$R_j = \frac{1}{N} \sum_{i=1}^N R_{ij}.$$

In our case, the number of samples, N , is the number of users in the test set, $N = 619,389$. The test compares the mean ranks of methods and is based on the following statistics:

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1) - \chi_F^2},$$

where

$$\chi_F^2 = \frac{12N}{K(K+1)} \sum_{i=1}^K R_i^2 - 3N(K+1)$$

is the Friedman statistic, which has the F distribution with $K-1$ and $(K-1)(N-1)$ degrees of freedom. The p -value from this test is equal to 0. The obtained p -value indicates that we can safely reject the null hypothesis that all algorithms perform the same. Therefore, we can proceed with the post hoc tests to detect significant pairwise differences among all the methods. Demšar [62] proposes the use of the Nemenyi test [63], which compares all algorithms pairwise. For a significance level α , the test determines the critical difference (CD). If the difference between the average rankings of two algorithms is greater than

$$CD = q_\alpha \sqrt{\frac{K(K+1)}{6N}},$$

the null hypothesis that the algorithms have the same performance is rejected (q_α is based on the studentized range statistic divided by $\sqrt{2}$). Demšar [62] proposed a plot to check the differences (the CD plot visually). Algorithms not joined by a line can be considered different in the plot.

In our case, with a significance of $\alpha = 0.05$, any two algorithms with a difference in the mean rank above 0.0114 will be regarded as non-equal (Figure 5).

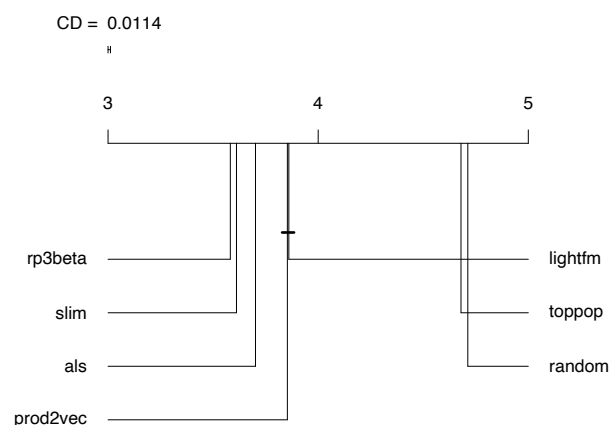


Figure 5. Critical difference plot (for precision). Source: own study.

5.1.3. Accuracy for Users with Different Numbers of Interactions

The only information we have about our users is their interactions. In Figure 6, we examine how the number of items a user has interacted with influences our main metric, precision@10. We divide our users into ten similarly sized groups based on the number of interactions.

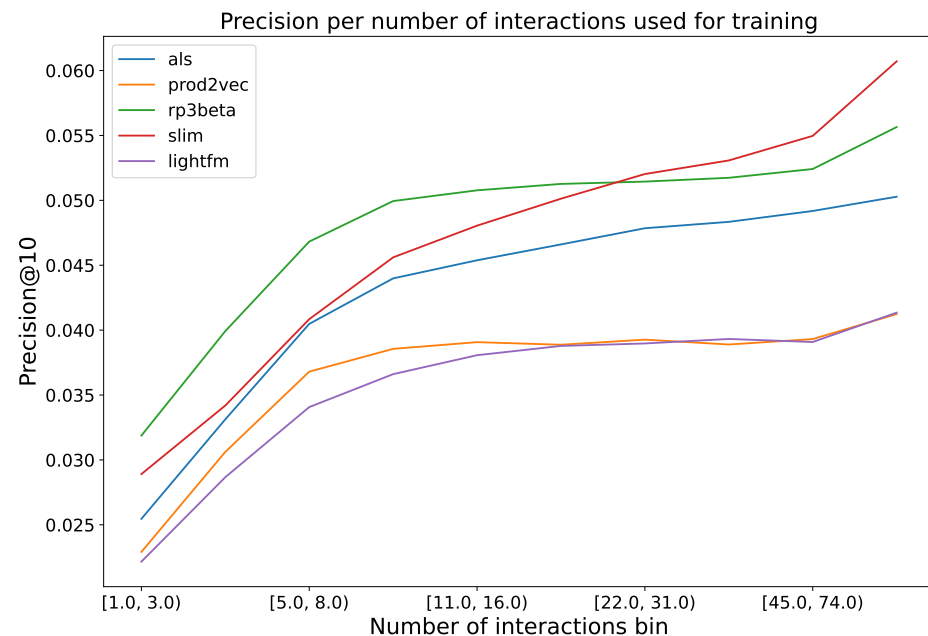


Figure 6. Precision@10 for each model, depending on the number of items with which the user interacted. It is best viewed in color. Source: own study.

We can see that the order of models sorted by precision@10 does not depend on the group, except that SLIM performed better than RP3Beta for users with at least 22 interactions. The difference is statistically significant for each group of users (discussed in Section 5.1.4). The RP3Beta model can be seen as a particular case of the item-based collaborative filtering approach, which uses a deterministic similarity measure between items [64]. On the other hand, SLIM uses machine learning methods to learn similarities. We suppose that such learned similarity scores are less biased but have higher variance, which leads to better performance for users with many interactions because averaging is done over a more significant number of scores.

5.1.4. Statistical Evaluation of Accuracy for Users with Different Numbers of Interactions

In this subsection, we discuss the difference between the RP3Beta and SLIM models depending on the number of user interactions, as observed in Figure 6.

To compare two methods over multiple datasets statistically, Demšar [62] recommends the Wilcoxon signed-rank test [65]. The Wilcoxon signed-rank test is a non-parametric alternative to the paired t test, which ranks the differences in performances of two algorithms for each dataset, ignoring the signs. It compares the ranks of the positive and negative differences.

Table 8 shows that the difference between the RP3Beta and SLIM models is statistically significant for each user group.

Table 8. Results of Wilcoxon signed-rank test between RP3Beta and SLIM models depending on the number of items with which the user interacted.

Bin	<i>p</i> -Value
[1.0, 3.0)	0
[3.0, 5.0)	0
[5.0, 8.0)	0
[8.0, 11.0)	0
[11.0, 16.0)	0
[16.0, 22.0)	1.25×10^{-6}
[22.0, 31.0)	9.27×10^{-7}
[31.0, 45.0)	0
[45.0, 74.0)	0
[74.0, 852.0)	0

Source: own study.

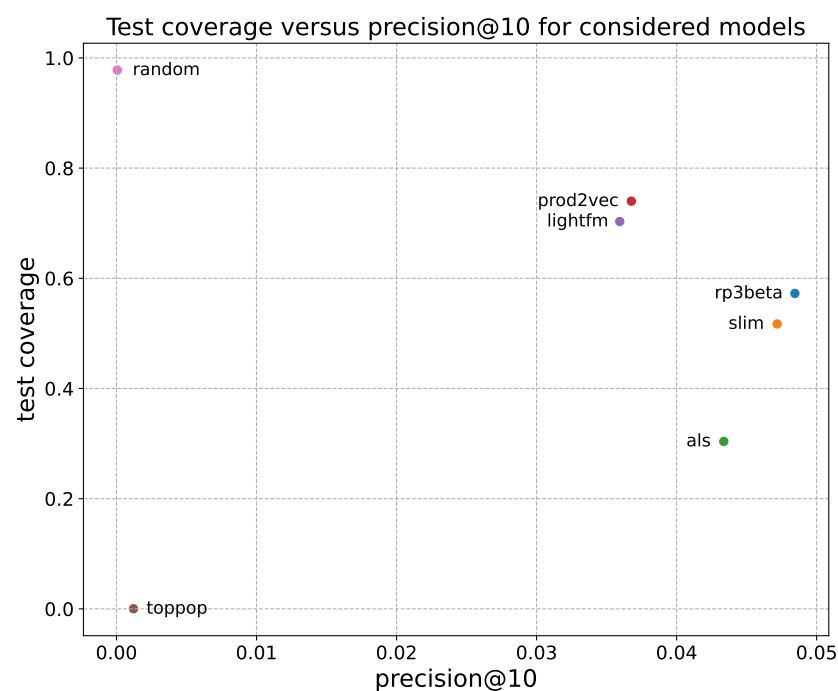
5.2. Diversity

In Table 9, we can observe that Prod2Vec and LightFM provide the greatest diversity after excluding random recommendations (which should have the greatest diversity). The diversities of the most accurate models, RP3Beta, SLIM, and ALS, are similar, except for the case of the lower test coverage of ALS. The relation between test coverage and precision@10 is shown in Figure 7.

Table 9. Diversity: evaluation results.

Metric	RP3Beta	SLIM	ALS	Prod2Vec	LightFM	Most Popular	Random
test coverage	0.5725	0.5171	0.3038	0.7400	0.7031	0.0002	0.9778
Shannon	9.5271	9.6728	9.6270	10.4031	10.1385	2.3296	11.7267
Gini	0.9083	0.9029	0.9120	0.7956	0.8397	0.9999	0.1159

Source: own study.

**Figure 7.** Test coverage versus precision@10 for considered models. Source: own study.

5.3. Overlap of Methods

In our research, we focus on methods that are worth testing with real users. Therefore, we studied the overlap between recommendations offered by the implemented methods concerning the user–item pairs (Overlap Coefficient [66]). We expect more significant differences in recommendations when comparing methods with a low overlap.

The results of this experiment are reported in Table 10. RP3Beta and SLIM offer similar recommendations. Although they are calculated differently, they use a sparse item similarity matrix. The comparison group with ALS has medium similarities with other methods. The most different from the overlap perspective are also the most diverse models, namely Prod2Vec and LightFM. The low overlap of ALS and LightFM emphasizes the importance of choosing the loss function in matrix factorization approaches.

Table 10. Overlap of models.

Model	RP3Beta	SLIM	ALS	Prod2Vec	LightFM
RP3Beta	100%	73%	53%	37%	38%
SLIM	73%	100%	50%	35%	35%
ALS	53%	50%	100%	38%	37%
Prod2Vec	37%	35%	38%	100%	28%
LightFM	38%	35%	37%	28%	100%

Source: own study.

5.4. Evaluation of Efficiency

The results of the efficiency evaluation are presented in Figure 8. RP3Beta, ALS, and LightFM outperform SLIM and Prod2Vec regarding execution time, whereas Prod2Vec requires the least memory. To retrain the model frequently, we set up the maximal acceptable execution time as 3 h, which was satisfied by ALS, LightFM, and RP3Beta. From a cost perspective, the acceptable memory utilization was set up to 110 GB, which all models fulfilled. We can observe that our models could be trained even on a single laptop.

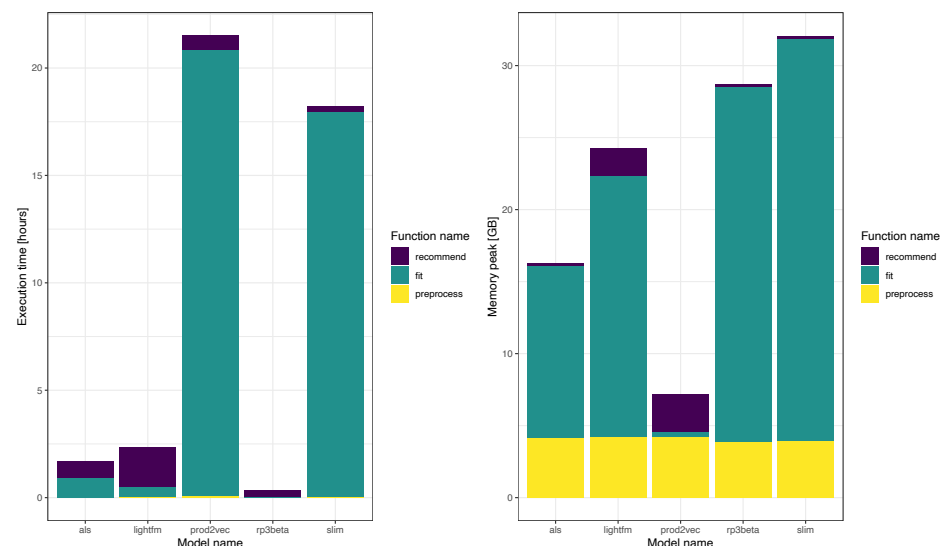


Figure 8. Execution time and memory utilization. It is best viewed in color. Source: own study.

5.5. Summary

The laboratory research enabled us to select models to be tested online with OLX users. Firstly, we chose the RP3Beta model for online tests because it is the best approach regarding accuracy and execution time. The relatively high memory consumption is still low enough for the model to be tested in production. Investigating the impact of this method's relatively low diversity may be an exciting line of future research.

SLIM is the second-best model in terms of precision@10. Nevertheless, we did not select this method for online evaluation due to its low efficiency and high overlap with the already selected RP3Beta model. Improving the efficiency of the SLIM implementation might change this decision.

Among other models, the highest precision@10 is achieved by ALS. Additionally, ALS was already implemented and used for job recommendations at OLX; therefore, we decided to include it in the online comparison.

Even though Prod2Vec and LightFM produce the most diverse recommendations, we decided not to test them online due to the significantly worse precision@10 and efficiency with Prod2Vec.

6. Online Evaluation

To evaluate the effectiveness of the selected methods, we conducted two online A/B tests with OLX Jobs users. We focused on users who had recently visited job ads on our platform. We sent them emails with ten recommended job ads, and when a user had installed an OLX application, a push notification with one recommended job ad was also sent. Then, we observed whether the user applied for any job within the next 48 h (converted user).

6.1. Impact of the ALS Model

The first experiment aimed to answer the question: does the introduction of the recommendation method increase the number of users applying for jobs? In this test, we split the users into two groups: a control group without recommendations and a group with recommendations generated using the ALS model. Only $\approx 10\%$ of users were assigned to the control group because the ALS model was already implemented and used for job recommendations at OLX before the test. We preferred to gather a minimal sample by running a more extended test to observe a longer-term impact on our users. The test was carried out for 25 days in March 2021. The results are reported in Table 11. The statistical significance of the advantage of ALS over the control group was examined using the chi-squared test, and the p -value was found to equal 0. This experiment confirmed our hypothesis that recommendations affect activity, as the percentage of converted users after receiving a communication increased significantly by $(16.83\% - 15.98\%)/15.98\% \approx 5.3\%$. The observed rise translates into hundreds of additional users applying for jobs through OLX Jobs daily.

Table 11. Comparison of activity with or without recommendations.

Variant	Users	% Converted Users
control	129,308	15.98%
ALS	1,170,262	16.83%

Source: own study.

6.2. Comparison of ALS and RP3Beta Models

In the second experiment, we tested whether the best model from laboratory testing outperformed ALS. We split our users into three groups, receiving recommendations from different recommendation systems: ALS, RP3Beta, and a mixed variant. We generated half of the recommendations with ALS and the other half with RP3Beta. The second experiment was conducted for 28 days in March and April 2021. The results are presented in Table 12. The percentage of converted users is calculated here relative to all users; however, changing the recommendation system did not affect users who did not open the message. We decided to filter these out, as their behavior added unnecessary noise to our data. Table 13 presents the impact on users who opened the message. The advantage of RP3Beta over ALS is statistically significant (p -value $\approx 10^{-6}$). The difference between the variants RP3Beta and ALS+RP3Beta is not statistically significant (the p -value is 0.19).

Table 12. Comparison of different recommendation methods.

Variant	Users	% Converted Users
ALS	343,892	15.25%
RP3Beta	345,273	15.40%
ALS+RP3Beta	343,896	15.30%

Source: own study.

Table 13. Comparison of different recommendation methods for users who opened the message.

Variant	Users	% Converted Users
ALS	44,775	19.66%
RP3Beta	46,097	20.94%
ALS+RP3Beta	45,469	20.59%

Source: own study.

Out of the users who opened the message, we observed 1.28 pp (percentage points) more converted users from RP3Beta than from ALS. Since the message open rate was 13.2%, this is equivalent to an increase of $1.28 \times 0.132 \approx 0.17$ pp of all targeted users. Based on the first experiment's results, ALS brings a 0.85 pp increase over the control group. This means that replacing ALS with RP3Beta increases our impact on all target users by approximately $0.17/0.85 = 20\%$.

6.3. Discussion

We have proved that sending job recommendations from the ALS model increases the number of users responding to job ads by more than 5%. Moreover, RP3Beta significantly outperforms ALS.

Additionally, we carried out another online A/B test comparing the RP3Beta model with a variant where we replaced half of the RP3Beta recommendations with recommendations generated from the deep learning hybrid recommender system utilized at OLX (<https://tech.olx.com/item2vec-neural-item-embeddings-to-enhance-recommendations-1fd948a6f293>, accessed on 29 May 2024). We decided not to describe that test in this paper, as this work focuses on collaborative filtering approaches. We also wanted to keep our results reproducible on the published dataset. Unfortunately, we could not publish all of the features utilized by our internal model. We did not observe any improvements over RP3Beta, even though the second model utilizes additional knowledge about job ads. Hence, we confirmed that a properly determined and tuned simple model might outperform more advanced deep solutions. This is consistent with the results from Dacrema et al. [64] and Anelli et al. [30], who compared several non-neural approaches, including RP3Beta, with more advanced neural models.

The efficiency of the RP3Beta model enables OLX Jobs to train it from scratch multiple times per day with very low costs (on a CPU). The state-of-the-art collaborative filtering model at OLX jobs is currently used to generate email and push recommendations for our users. We recently adapted this model to serve on-site recommendations in real-time, considering recent user interactions with items [67]. We also proposed the P3 learning to rank (P3LTR) model [68], a generalization of the RP3Beta, where we learn the importance of user–item relations based on types of user–item past interactions and timestamps.

In this work, we have addressed the research gap of the lack of comparisons of classical collaborative filtering approaches in the job domain by providing comprehensive laboratory research on a real-world dataset and an online evaluation with millions of users. Additionally, we have made the source code for offline assessment and the dataset publicly available. We encourage other researchers to develop new recommendation methods to help users find their ideal job more quickly.

The methods considered in this work only require (user, item, and timestamp) triplets as input data for training and evaluation; hence, they can be applied across several domains,

even for datasets consisting of tens of millions of interactions. We expect good results in the classifieds domain based on our initial results obtained in other categories of OLX.

7. Summary

This paper has addressed the topic of job recommendations in online classifieds, using the example of OLX Jobs. We performed an extensive evaluation of approaches representing different families of recommendation methods.

During the laboratory research, we demonstrated that RP3Beta, SLIM, and ALS perform significantly better than LightFM and Prod2Vec. We also found that Prod2Vec offers the most extraordinary diversity, which may be necessary when working on the novelty of recommendations. Moreover, we examined how similar the recommendations generated by different models are.

We also conducted an evaluation by sending millions of messages to online users. A/B tests were carried out using ALS and RP3Beta. The results of these tests demonstrated that sending job recommendations generated by these models increases the number of users contacting advertisers from OLX Jobs. With RP3Beta available in the production environment, we improved our effectiveness by 20% (when considering user reactions).

When evaluating the methods from the classifieds perspective, a model recommending the most popular (as yet unseen) items was unsuitable for job recommendations. A recommendation model based on random recommendations was also added for comparison to demonstrate the advantage of the researched methods.

Another significant result of this research is the dataset, which will enable further research on recommendations for classified needs.

One of the weaknesses of our work is a lack of comparison against deep neural networks (DNNs) or graph neural networks (GNNs). For such methods, satisfying our time constraints without more expensive infrastructure (e.g., with GPU) is challenging. Relaxing our constraints or comparing the most efficient DNNs or GNNs would be interesting for future work. Another weakness is that the comparison is only on a single dataset. The size of the dataset has a significant impact on method selection. It would be interesting to benchmark the same methods on datasets of different sizes, either subsets of the original dataset or other datasets from the job domain (e.g., from different markets the OLX operates). Additionally, it would be interesting to explore the correlation between diversity and online results, aiming to develop an offline metric that better aligns with business goals.

Author Contributions: Conceptualization, R.K.; methodology, R.K. and T.G.; software, R.K. and V.D.; validation, R.K., V.D., T.G. and A.F.; formal analysis, R.K. and T.G.; investigation, R.K. and T.G.; resources, R.K.; data curation, R.K. and V.D.; writing—original draft preparation, R.K., A.F. and T.G.; writing—review and editing, R.K., A.F. and T.G.; visualization, R.K. and T.G.; supervision, T.G. and A.F.; project administration, R.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Dataset: <https://www.kaggle.com/datasets/olxdatascience/olx-jobs-interactions>, accessed on 29 May 2024; source code: <https://github.com/rob-kwiec/olx-jobs-recommendations>, accessed on 29 May 2024

Conflicts of Interest: Authors Robert Kwieciński, Agata Filipowska and Viacheslav Dubrov were employed by the company OLX Group. The remaining author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Davis, M. What are Online Classifieds? *Big Easy Magazine*, 4 January 2021.
2. Jones, S. *Online Classifieds*; Pew Internet & American Life Project: Washington, DC, USA, 2009.
3. Gourville, J.; Soman, D. Overchoice and Assortment Type: When and Why Variety Backfires. *Mark. Sci.* **2005**, *24*, 382–395. [CrossRef]

4. Chernev, A. When More Is Less and Less Is More: The Role of Ideal Point Availability and Assortment in Consumer Choice. *J. Consum. Res.* **2003**, *30*, 170–183. [\[CrossRef\]](#)
5. Boatwright, P.; Nunes, J. Reducing Assortment: An Attribute-Based Approach. *J. Mark.* **2001**, *65*, 50–63. [\[CrossRef\]](#)
6. Gómez-Urbe, C.; Hunt, N. The Netflix Recommender System. *ACM Trans. Manag. Inf. Syst.* **2015**, *6*, 1–19. [\[CrossRef\]](#)
7. Smith, B.; Linden, G. Two Decades of Recommender Systems at Amazon.com. *IEEE Internet Comput.* **2017**, *21*, 12–18. [\[CrossRef\]](#)
8. Aggarwal, C.C. *Recommender Systems: The Textbook*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2016.
9. Li, S.; Karahanna, E. Online Recommendation Systems in a B2C E-Commerce Context: A Review and Future Directions. *J. Assoc. Inf. Syst.* **2015**, *16*, 72–107. [\[CrossRef\]](#)
10. Twardowski, B. Modelling Contextual Information in Session-Aware Recommender Systems with Neural Networks. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 273–276. [\[CrossRef\]](#)
11. Alotaibi, S. A survey of job recommender systems. *Int. J. Phys. Sci.* **2012**, *7*, 5127–5142. [\[CrossRef\]](#)
12. Abel, F.; Benczúr, A.; Kohlsdorf, D.; Larson, M.; Pálovics, R. *RecSys Challenge'16: Proceedings of the Recommender Systems Challenge*; Association for Computing Machinery: New York, NY, USA, 2016.
13. Abel, F.; Deldjoo, Y.; Elahi, M.; Kohlsdorf, D. *RecSys Challenge'17: Proceedings of the Recommender Systems Challenge 2017*; Association for Computing Machinery: New York, NY, USA, 2017.
14. Li, S.; Shi, B.; Yang, J.; Yan, J.; Wang, S.; Chen, F.; He, Q. Deep Job Understanding at LinkedIn. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 25–30 July 2020; pp. 2145–2148. [\[CrossRef\]](#)
15. Zhu, G.; Chen, Y.; Wang, S. Graph-Community-Enabled Personalized Course-Job Recommendations with Cross-Domain Data Integration. *Sustainability* **2022**, *14*, 7439. [\[CrossRef\]](#)
16. Lacic, E.; Reiter-Haas, M.; Kowald, D.; Reddy Dareddy, M.; Cho, J.; Lex, E. Using autoencoders for session-based job recommendations. *User Model. User-Adapt. Interact.* **2020**, *30*, 617–658. [\[CrossRef\]](#)
17. Pazzani, M.J.; Billsus, D. Content-Based Recommendation Systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 325–341. [\[CrossRef\]](#)
18. Chen, R.; Hua, Q.; Chang, Y.S.; Wang, B.; Zhang, L.; Kong, X. A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks. *IEEE Access* **2018**, *6*, 64301–64320. [\[CrossRef\]](#)
19. Wang, X.; He, X.; Wang, M.; Feng, F.; Chua, T.S. Neural Graph Collaborative Filtering. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 165–174. [\[CrossRef\]](#)
20. Schafer, J.B.; Frankowski, D.; Herlocker, J.; Sen, S. Collaborative filtering recommender systems. In *The Adaptive Web*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 291–324. [\[CrossRef\]](#)
21. Takács, G.; Pilászy, I.; Németh, B.; Tikk, D. Scalable Collaborative Filtering Approaches for Large Recommender Systems. *J. Mach. Learn. Res.* **2009**, *10*, 623–656. [\[CrossRef\]](#)
22. Jawaheer, G.; Szomszor, M.; Kostkova, P. Comparison of implicit and explicit feedback from an online music recommendation service. In Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, Barcelona, Spain, 26 September 2010; pp. 47–51. [\[CrossRef\]](#)
23. Tamm, Y.M.; Damdinov, R.; Vasilev, A. Quality Metrics in Recommender Systems: Do We Calculate Metrics Consistently? In Proceedings of the Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September–1 October 2021; pp. 708–713. [\[CrossRef\]](#)
24. Kaminskas, M.; Bridge, D. Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems. *ACM Trans. Interact. Intell. Syst.* **2016**, *7*, 1–42. [\[CrossRef\]](#)
25. Kunaver, M.; Pozrl, T. Diversity in Recommender Systems, A Survey. *Knowl.-Based Syst.* **2017**, *123*, 154–162. [\[CrossRef\]](#)
26. Ge, M.; Delgado-Battenfeld, C.; Jannach, D. Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity. In Proceedings of the Fourth ACM Conference on Recommender Systems, Barcelona, Spain, 26–30 September 2010; pp. 257–260. [\[CrossRef\]](#)
27. Mogenet, A.; Pham, T.A.N.; Kazama, M.; Kong, J. Predicting Online Performance of Job Recommender Systems with Offline Evaluation. In Proceedings of the 13th ACM Conference on Recommender Systems, Copenhagen, Denmark, 16–20 September 2019; pp. 477–480. [\[CrossRef\]](#)
28. Anelli, V.W.; Bellogín, A.; Di Noia, T.; Jannach, D.; Pomo, C. Top-N Recommendation Algorithms: A Quest for the State-of-the-Art. In Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization, Barcelona, Spain, 4–7 July 2022; pp. 121–131. [\[CrossRef\]](#)
29. Dacrema, M.F.; Boglio, S.; Cremonesi, P.; Jannach, D. A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research. *ACM Trans. Inf. Syst.* **2021**, *39*, 1–49. [\[CrossRef\]](#)
30. Anelli, V.W.; Bellogín, A.; Di Noia, T.; Pomo, C. Reenvisioning the Comparison between Neural Collaborative Filtering and Matrix Factorization. In Proceedings of the Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September–1 October 2021; pp. 521–529. [\[CrossRef\]](#)

31. Zhu, J.; Dai, Q.; Su, L.; Ma, R.; Liu, J.; Cai, G.; Xiao, X.; Zhang, R. BARS: Towards Open Benchmarking for Recommender Systems. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 2912–2923. [\[CrossRef\]](#)
32. Dong, Y.; Li, J.; Schnabel, T. When Newer is Not Better: Does Deep Learning Really Benefit Recommendation From Implicit Feedback? In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, Taipei, Taiwan, 23–27 July 2023; pp. 942–952. [\[CrossRef\]](#)
33. Anelli, V.W.; Malitesta, D.; Pomo, C.; Bellogin, A.; Di Sciascio, E.; Di Noia, T. Challenging the Myth of Graph Collaborative Filtering: A Reasoned and Reproducibility-driven Analysis. In Proceedings of the 17th ACM Conference on Recommender Systems, Singapore, 18–22 September 2023; pp. 350–361. [\[CrossRef\]](#)
34. Koren, Y.; Bell, R.; Volinsky, C. Matrix Factorization Techniques for Recommender Systems. *Computer* **2009**, *42*, 30–37. [\[CrossRef\]](#)
35. Kula, M. Metadata embeddings for user and item cold-start recommendations. *arXiv* **2015**, arXiv:1507.08439.
36. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian Personalized Ranking from Implicit Feedback. *arXiv* **2012**, arXiv:1205.2618
37. Weston, J.; Bengio, S.; Usunier, N. Wsabie: Scaling up to large vocabulary image annotation. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011; pp. 2764–2770.
38. Weston, J.; Yee, H.; Weiss, R.J. Learning to Rank Recommendations with the K-Order Statistic Loss. In Proceedings of the 7th ACM Conference on Recommender Systems, Hong Kong, China, 12–16 October 2013; pp. 245–248. [\[CrossRef\]](#)
39. Hu, Y.; Koren, Y.; Volinsky, C. Collaborative Filtering for Implicit Feedback Datasets. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 263–272. [\[CrossRef\]](#)
40. Takács, G.; Pilászy, I.; Tikk, D. Applications of the Conjugate Gradient Method for Implicit Feedback Collaborative Filtering. In Proceedings of the Fifth ACM Conference on Recommender Systems, Chicago, IL, USA, 23–27 October 2011; pp. 297–300. [\[CrossRef\]](#)
41. Ning, X.; Karypis, G. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In Proceedings of the 2011 IEEE 11th International Conference on Data Mining, Vancouver, BC, Canada, 11–14 December 2011; pp. 497–506. [\[CrossRef\]](#)
42. Christoffel, F.; Paudel, B.; Newell, C.; Bernstein, A. Blockbusters and Wallflowers: Accurate, Diverse, and Scalable Recommendations with Random Walks. In Proceedings of the 9th ACM Conference on Recommender Systems, Vienna, Austria, 16–20 September 2015; pp. 163–170. [\[CrossRef\]](#)
43. Cooper, C.; Lee, S.H.; Radzik, T.; Siantos, Y. Random Walks in Recommender Systems: Exact Computation and Simulations. In Proceedings of the 23rd International Conference on World Wide Web, New York, NY, USA, 7–11 April 2014; pp. 811–816. [\[CrossRef\]](#)
44. Grbovic, M.; Radosavljevic, V.; Djuric, N.; Bhamidipati, N.; Savla, J.; Bhagwan, V.; Sharp, D. E-Commerce in Your Inbox: Product Recommendations at Scale. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, 10–13 August 2015; pp. 1809–1818. [\[CrossRef\]](#)
45. Barkan, O.; Koenigstein, N. ITEM2VEC: Neural item embedding for collaborative filtering. In Proceedings of the 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), Vietri sul Mare, Italy, 13–16 September 2016; pp. 1–6. [\[CrossRef\]](#)
46. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems—Volume 2, Red Hook, NY, USA, 5–10 December 2013; pp. 3111–3119.
47. De Ruijt, C.; Bhulai, S. Job recommender systems: A review. *arXiv* **2021**, arXiv:2111.13576.
48. Kula, M. LightFM—Github Repository. 2023. Available online: <https://github.com/lyst/lightfm> (accessed on 29 May 2024).
49. Frederickson, B. ALS—Github Repository. 2022. Available online: <https://github.com/benfred/implicit/blob/main/implicit/als.py> (accessed on 29 May 2024).
50. Levy, M.; Grisel, O. SLIM—Github Repository. 2014. Available online: https://github.com/Mendeley/mrec/blob/master/mrec/item_similarity/slim.py (accessed on 29 May 2024).
51. Quadrana, M. SLIM—Github Repository. 2021. Available online: https://github.com/MaurizioFD/RecSys2019_DeepLearning_Evaluation/blob/master/SLIM_ElasticNet/SLIMElasticNetRecommender.py (accessed on 29 May 2024).
52. Bernardis, C. RP3Beta—Github Repository. 2021. Available online: https://github.com/MaurizioFD/RecSys2019_DeepLearning_Evaluation/blob/master/GraphBased/RP3betaRecommender.py (accessed on 29 May 2024).
53. Paudel, B.; Christoffel, F.; Newell, C.; Bernstein, A. Updatable, Accurate, Diverse, and Scalable Recommendations for Interactive Applications. *ACM Trans. Interact. Intell. Syst.* **2016**, *7*, 1–34. [\[CrossRef\]](#)
54. Řehůřek, R.; Sojka, P. Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, 22 May 2010; pp. 45–50, ISBN 2-9517408-6-7. Available online: <http://is.muni.cz/publication/884893/en> (accessed on 29 May 2012).
55. Meng, Z.; McCreddie, R.; Macdonald, C.; Ounis, I. Exploring Data Splitting Strategies for the Evaluation of Recommendation Models. In Proceedings of the 14th ACM Conference on Recommender Systems, Virtual, 22–26 September 2020; pp. 681–686. [\[CrossRef\]](#)
56. Shani, G.; Gunawardana, A., Evaluating Recommendation Systems. In *Recommender Systems Handbook*; Springer: Boston, MA, USA, 2011; pp. 257–297. [\[CrossRef\]](#)

57. Friedman, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [[CrossRef](#)]
58. Friedman, M. A Comparison of Alternative Tests of Significance for the Problem of m Rankings. *Ann. Math. Stat.* **1940**, *11*, 86–92. [[CrossRef](#)]
59. Iman, R.L.; Davenport, J.M. Approximations of the critical region of the Friedman statistic. *Commun. Stat.-Theory Methods* **1980**, *9*, 571–595. [[CrossRef](#)]
60. García, S.; Herrera, F. An extension on “Statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons. *J. Mach. Learn. Res.* **2008**, *9*, 2677–2694.
61. García, S.; Fernández, A.; Luengo, J.; Herrera, F. Advanced Nonparametric Tests for Multiple Comparisons in the Design of Experiments in Computational Intelligence and Data Mining: Experimental Analysis of Power. *Inf. Sci.* **2010**, *180*, 2044–2064. [[CrossRef](#)]
62. Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
63. Nemenyi, P. Distribution-Free Multiple Comparisons. Ph.D. Thesis, Princeton University, Princeton, NJ, USA, 1963.
64. Dacrema, M.F.; Cremonesi, P.; Jannach, D. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. In Proceedings of the 13th ACM Conference on Recommender Systems, Copenhagen, Denmark, 16–20 September 2019; pp. 101–109. [[CrossRef](#)]
65. Wilcoxon, F. Individual Comparisons by Ranking Methods. *Biom. Bull.* **1945**, *1*, 80–83. [[CrossRef](#)]
66. Vijaymeena, M.; Kavitha, K. A Survey on Similarity Measures in Text Mining. *Mach. Learn. Appl. Int. J.* **2016**, *3*, 19–28. [[CrossRef](#)]
67. Kwieciński, R.; Melniczak, G.; Górecki, T. Comparison of Real-Time and Batch Job Recommendations. *IEEE Access* **2023**, *11*, 20553–20559. [[CrossRef](#)]
68. Kwieciński, R.; Górecki, T.; Filipowska, A. Learning edge importance in bipartite graph-based recommendations. In Proceedings of the 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS), Sofia, Bulgaria, 4–7 September 2022; pp. 227–233. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Efficient Speech Signal Dimensionality Reduction Using Complex-Valued Techniques

Sungkyun Ko ¹  and Minho Park ^{2,*} ¹ Department of AI IT Convergence, Soongsil University, Seoul 06978, Republic of Korea; kosungkyun@naver.com² School of Electronic Engineering, Soongsil University, Seoul 06978, Republic of Korea

* Correspondence: mhp@ssu.ac.kr

Abstract: In this study, we propose the CVMFCC-DR (Complex-Valued Mel-Frequency Cepstral Coefficients Dimensionality Reduction) algorithm as an efficient method for reducing the dimensionality of speech signals. By utilizing the complex-valued MFCC technique, which considers both real and imaginary components, our algorithm enables dimensionality reduction without information loss while decreasing computational costs. The efficacy of the proposed algorithm is validated through experiments which demonstrate its effectiveness in building a speech recognition model using a complex-valued neural network. Additionally, a complex-valued softmax interpretation method for complex numbers is introduced. The experimental results indicate that the approach yields enhanced performance compared to traditional MFCC-based techniques, thereby highlighting its potential in the field of speech recognition.

Keywords: mel-frequency cepstral coefficients; complex-valued neural network error correction; backpropagation; complex-valued softmax



Citation: Ko, S.; Park, M. Efficient Speech Signal Dimensionality Reduction Using Complex-Valued Techniques. *Electronics* **2024**, *13*, 3046. <https://doi.org/10.3390/electronics13153046>

Academic Editors: Xin Geng, Padma Iyengar and Elke Pulvermüller

Received: 3 June 2024

Revised: 31 July 2024

Accepted: 31 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Today, speech signal processing and recognition play crucial roles in various application domains. These technologies are utilized for diverse purposes, such as converting speech commands into text or identifying the owner of a voice. The advancement of speech recognition technology can improve human–machine interaction, thereby enhancing productivity and efficiency across various industries. However, speech signals possess highly diverse forms and high-dimensional characteristics, making effective processing and recognition a persistent challenge. The diversity and high dimensionality of speech signals make dimensionality reduction and pattern recognition difficult, leading to increased computational costs. Therefore, efficiently reducing the dimensionality of speech signals and improving the performance of neural network-based speech recognition models is a significant challenge.

Speech processing and recognition technologies are essential in various applications, such as voice interfaces, virtual assistants, and automatic translation. Achieving more accurate and efficient speech signal processing and recognition results can significantly enhance the user experience of related services. Consequently, continuous research is being conducted in the field of speech signal processing and recognition. Speech signals exhibit high-dimensional characteristics, making effective dimensionality reduction challenging. Moreover, accurate recognition of diverse speech patterns requires intricate neural network models, which, in turn, increases computational costs [1]. Therefore, effectively reducing the dimensionality of speech signals without information loss and simultaneously enhancing the accuracy and efficiency of neural network-based speech recognition models is a crucial task.

Despite the rich literature on complex-valued networks, there have been few examples of their application in audio or speech processing. Given that the human auditory system

processes sound in the frequency domain, there is also meaning in exploring complex-valued neural networks for audio processing rather than relying solely on real-valued neural networks in the time domain [2].

Recent research has made significant progress in speech recognition by combining traditional methods like mel-frequency cepstral coefficients (MFCCs) with neural networks [3]. While the MFCC technique has been widely used for speech feature extraction, it has the drawback of being unable to perform lossless dimensionality reduction, as it uses only real values. In other words, reducing the number of MFCCs leads to information loss, which decreases speech recognition accuracy. These methods still exhibit limited effectiveness, and improved approaches beyond traditional learning methods are necessary.

Utilizing the complex representation of MFCC allows for the consideration of both real and imaginary parts, enabling lossless dimensionality reduction. Additionally, employing complex-valued neural networks can reflect phase changes through complex multiplication operations [4], better representing the characteristics of speech signals. This approach is expected to enable more efficient dimensionality reduction and neural network-based speech recognition compared to existing methods.

In this study, we propose the CVMFCC-DR algorithm for efficient and lossless speech signal dimensionality reduction. By combining CVMFCC-DR with complex-valued neural networks, this technique can demonstrate superior speech recognition performance compared to existing methods.

The main contributions of this study are as follows:

- Proposing an efficient dimensionality reduction technique (CVMFCC-DR) utilizing the complex representation of MFCCs.
- Enhancing speech recognition performance by combining complex-valued MFCCs and complex-valued neural networks.
- Proposing a new interpretation method for a complex-valued softmax function with complex inputs.

This paper begins by introducing the existing MFCC technique and complex-valued neural networks. It then provides a detailed explanation of the proposed CVMFCC-DR algorithm and the learning method for complex-valued neural networks [5]. Subsequently, the effectiveness of the proposed method is experimentally validated, and the results are analyzed. Finally, the paper concludes with a summary of the study's findings and suggests directions for future research.

2. Related Works

In the field of speech recognition, various neural network-based approaches and techniques are utilized. Different types of neural networks, such as CNNs, RNNs, LSTM, and DNNs, are used, each with their own advantages and disadvantages [3]. Moreover, hybrid models that combine traditional methods with neural network structures are also being researched. Hybrid models can optimize performance by combining the strengths of each structure, but they also come with the drawback of increased complexity in implementation and interpretation [6].

Amidst these developments, traditional feature extraction techniques like the MFCC technique are still widely used as inputs for neural networks [3]. Additionally, exploring novel neural architectures such as complex-valued neural networks could uncover new perspectives and potential breakthroughs in speech recognition capabilities. Recently, there have been studies showing that FPGAs can be used to significantly improve complex number computation speed [7]. These research results empirically demonstrate that the hardware implementation of complex number operations can bring about practical performance improvements. Furthermore, a recent study implemented complex-valued neural networks on FPGAs for classification, demonstrating superior accuracy compared to real-valued neural networks and achieving significant speed improvements over CPU-based systems [8]. This gives rise to the expectation that in the future, CPUs that currently only support real number operations may one day support complex number operations as well.

Based on these studies, utilizing complex number representations enables the encoding of real number pairs as single complex numbers, effectively reducing the input dimension by half. This reduction in dimensionality decreases the number of model parameters and enhances computational efficiency. Consequently, experimental evidence shows that complex-valued neural networks can achieve equal or superior performance with fewer parameters compared to conventional real-valued neural networks.

As research continues to push the boundaries, a synergistic combination of proven methods and innovative approaches may pave the way for the next generation of robust and high-performing speech recognition systems.

3. Background Knowledge

Various studies are being conducted in the field of speech recognition and classification, and in this section, we introduce MFCCs (mel-frequency cepstral coefficients) [3]. The MFCC technique is one of the feature extraction methods commonly used in the fields of speech signal processing and speech recognition. The MFCC technique represents the frequency characteristics of speech signals and is primarily utilized to effectively represent and recognize speech signals. The basic principle of the MFCC technique is illustrated in Figure 1.



Figure 1. An overview of the process of the MFCC technique.

Pre-emphasis: Pre-emphasis is essentially a high-pass filter. Due to the structure of the human body, high-frequency components tend to decrease in the actual emitted sound during speech production. Therefore, pre-emphasis is the process of applying a high-pass filter to amplify the attenuated high-frequency components first.

Framing and Windowing: Speech signals are divided into short segments called frames for analysis. This is because speech characteristics remain relatively stable over brief periods. These frames slightly overlap to capture speech changes more effectively. Each frame is processed using a window technique, which enhances analysis. Common window types include Hamming and Hanning windows. This framing method allows for more accurate analysis of complex speech data, which is crucial for developing speech recognition and synthesis technologies. While specific implementation details may vary, these fundamental principles are widely used in speech processing systems.

Discrete Fourier Transform: For calculating the power spectrum, DFT (Discrete Fourier Transform) is used to compute the power spectrum for each frame obtained earlier. This process enables the discernment of the frequency-specific energy distribution of the speech signal across each frame.

Mel-Frequency Filter Bank: The mel band-pass filter is a set of filters designed to extract speech signals in a nonlinear manner, mimicking the human auditory system. Typically constructed with 40 triangular filters, it analyzes the frequency characteristics of speech signals.

Discrete Cosine Transform: DCT (Discrete Cosine Transform) expresses a finite sequence of data points as a sum of cosine functions oscillating at different frequencies. In the MFCC process, DCT is applied to the mel filter bank to select acceleration coefficients or to separate the relationship in log-spectral magnitudes from the filter bank.

Speech recognition using MFCCs can be effectively employed in various environments [9], and this method can achieve high accuracy when combined with deep learning models. Therefore, MFCC serves as a crucial input for deep learning models for speech recognition [10].

4. Proposed Method

In this section, we propose a method for efficiently performing speech signal dimensionality reduction using complex-valued MFCCs and complex-valued neural networks. An overview of the proposed approach is presented in Figure 2. This figure illustrates the proposed approach for voice classification using complex-valued neural networks. The process commences with voice input, followed by feature extraction using mel-frequency cepstral coefficients (MFCCs). These features are then subjected to dimensionality reduction and transformed into complex-valued representations, resulting in complex-valued MFCCs. The processed features are subsequently inputted into a complex-valued neural network. The network's performance is enhanced through the selection of appropriate activation functions and learning algorithms. Ultimately, the neural network's output is utilized for classification purposes.

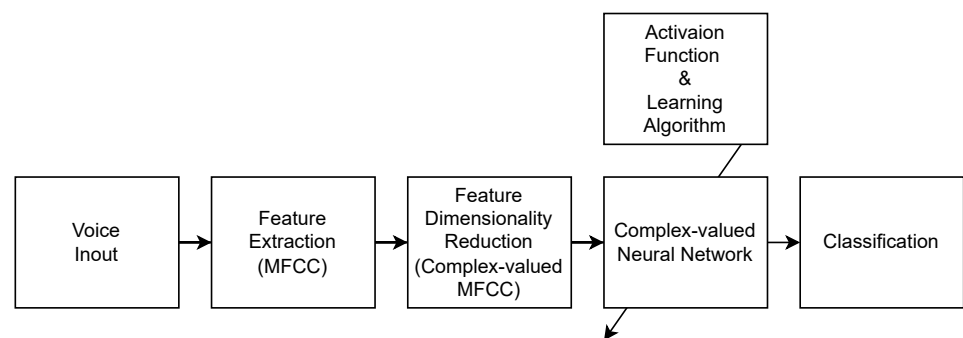


Figure 2. Overview of the proposed approach.

Complex-valued MFCCs: The number of MFCCs used in speech recognition can vary depending on specific applications and models. However, typically, MFCCs ranging from 13 to 40 are used. This range is known to effectively operate in various speech recognition systems. Generally, a higher number of MFCCs can allow the model to capture more frequency and spectral detail, but it also increases the computational cost of the model. Therefore, the selected number of MFCCs is determined considering a balance between computational efficiency and model performance.

When representing physical problems using complex numbers, there are two main methods. One is a simple substitution method, and the other is a more elegant approach known as mathematical equivalence, like the Fourier Transform. Substitution takes two real physical parameters and places one in the real part of the complex number and one in the imaginary part. This allows the two values to be manipulated as a single entity, i.e., a single complex number [11]. The method proposed for dimensionality reduction in this paper involves converting n MFCC values into complex numbers. The first MFCC is used as the real part, and the second MFCC is used as the imaginary part of the complex number. Subsequently, each MFCC alternates between being the real and imaginary parts within the complex number. This pattern continues up to the n th MFCC, resulting in $n/2$ complex numbers. For example, when we have n MFCC values, they could be as follows:

$$[x_1, x_2, x_3, x_4, \dots, x_{n-1}, x_n]$$

In this case, the first MFCC x_1 is used as the real part of the first complex number. The second MFCC x_2 is used as the imaginary part of the first complex number. The third MFCC x_3 is used as the real part of the second complex number. The fourth MFCC x_4 is used as the imaginary part of the second complex number. This pattern continues alternating between real and imaginary parts up to the n th MFCC x_n , which becomes the imaginary part of the $(n/2)$ -th complex number. Therefore, with n MFCC values, $n/2$ complex numbers are generated. The first complex number is $x_1 + jx_2$, the second complex

number is $x_3 + jx_4$, and so on until the last complex number, which is $x_{n-1} + jx_n$. We have illustrated this explanation in Figure 3.

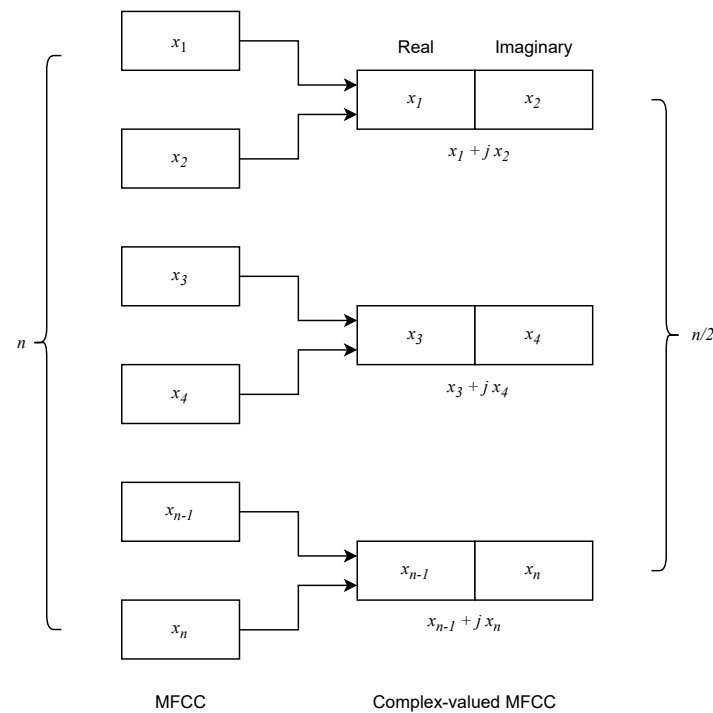


Figure 3. Process of generating complex-valued MFCCs.

Complex-Valued Neural Network: Complex-valued neural networks are neural network models that use complex numbers as input values or weights, and they have the advantage of handling richer information compared to real-valued neural networks. Furthermore, various research studies have been conducted in various fields, and complex-valued dimensionality reduction methods or classification and detection using complex numbers are being effectively utilized. For this purpose, the multiplication of complex inputs and complex weights is performed, which implies rotation in the complex plane and thus introduces changes in phase [5]. The sum of the multiplication of input X and weight W can be represented as Equations (1) and (2).

$$Z = W_0 + W_1X_1 + W_2X_2 + \dots + W_nX_n \quad (1)$$

$$Z = \sum_n W_n \cdot X_n \quad (2)$$

Activation Function and Learning Algorithm: In complex-valued neural networks, a phase-dependent activation function is used, forcing the magnitude information to be normalized to a unit circle of 1 and only the extraction of the phase information, as shown in Equation (3).

$$P(Z) = \frac{Z}{|Z|} = e^{j \cdot \arg(Z)} \quad (3)$$

The learning algorithm of complex-valued neural networks is as follows. When there is a desired target t and an output value z , as shown in Figure 4, the weights are adjusted according to Equation (4) to gradually move in the direction of the desired target [12].

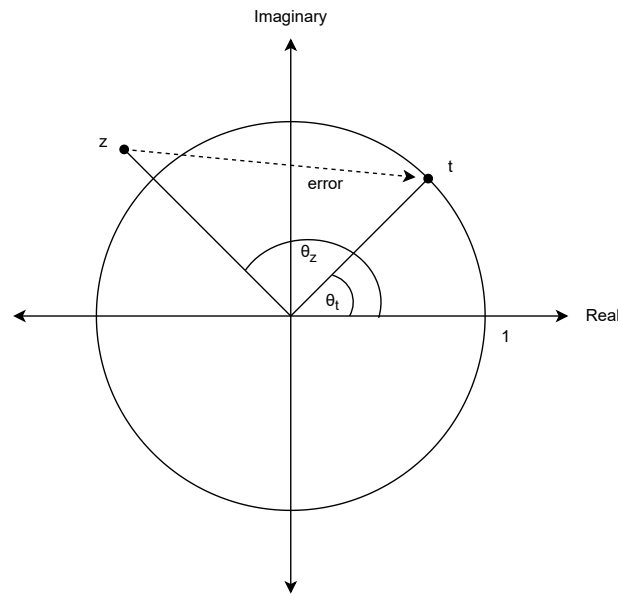


Figure 4. Learning algorithm for CVNN.

$$t = \sum_n (W_n + \Delta W_n) \cdot X_n \quad (4)$$

The error with respect to the training target t is $e = t - z$, as shown in Equation (5).

$$e = \sum_n (W_n + \Delta W_n) \cdot X_n - \sum_n (W_n) \cdot X_n = \sum_n (\Delta W_n) \cdot X_n \quad (5)$$

To simplify and expedite the calculation of the error with respect to weight changes, assuming that each node contributes equally to the error, the summation symbol σ is removed from the equation, and both sides are divided by the number of contributing nodes N in the complex-valued neural network for computational convenience. The resulting equation is Equation (6).

$$\Delta W_n \cdot X_n = \frac{e}{N} \quad (6)$$

To obtain ΔW_n in order to change the weights, multiply both sides of Equation (6) by X_n^{-1} . X_n^{-1} becomes $\overline{X_n}$ when multiplied by the complex conjugate $\overline{X_n}$, with n being both the numerator and denominator; thus, $X_n^{-1} = \frac{\overline{X_n}}{|X_n|^2}$, where $|X_n| = 1$ for X_n within the unit circle. Therefore, $X_n^{-1} = \overline{X_n}$, and Equation (6) is derived again as Equations (7) and (8).

$$\Delta W_n \cdot X_n \cdot X_n^{-1} = \frac{e}{N} \cdot X_n^{-1} = \frac{e}{N} \cdot \overline{X_n} \quad (7)$$

$$\Delta W_n = \frac{e}{N} \cdot \overline{X_n} \quad (8)$$

In this way, the obtained change in weights can be observed in Equation (8). This change represents the adjustment of complex weights for each node contributing to the error with respect to the training target and is uniformly applied across the entire network. Through this process, complex-valued neural networks adjust closer to the target utilizing phase-dependent activation functions and efficient learning algorithms while handling rich information through complex multiplication operations between inputs and weights [13].

Complex Domain Backpropagation: When using hidden layers, the error from the output layer is backpropagated to the hidden layers to calculate how much each node

in the hidden layers contributed to that error. Based on this, the weights between the hidden and output layers need to be adjusted appropriately. This process is called error backpropagation, through which the neural network can adjust the weights to minimize the error in the output values during learning. The structure of a multi-layer complex-valued neural network is illustrated in Figure 5.

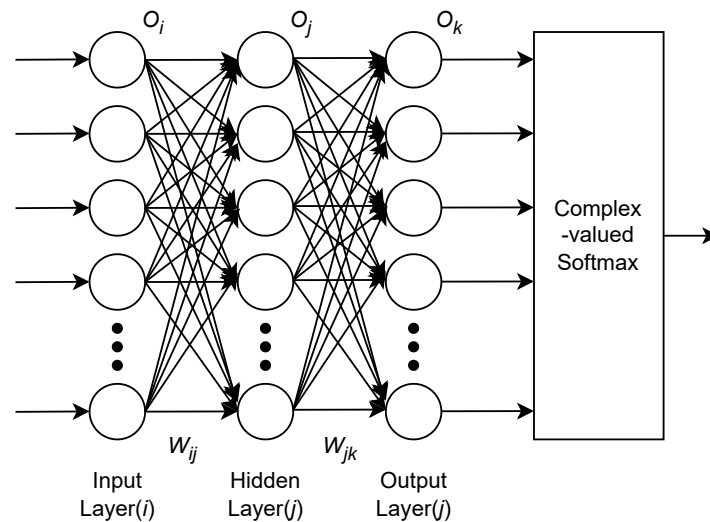


Figure 5. Multi-layer CVNN.

In this paper, an error correction method is used for the output layer k to update the weights, while an error backpropagation method is employed for the hidden layer j to examine its influence on the output. In other words, the output layer k utilizes the error correction method to avoid falling into local minima, while the error backpropagation method is applied to the hidden layer j to ensure better distribution of errors from the output layer to the hidden layer. This combination of error backpropagation in the hidden layer and error correction in the output layer can be expected to improve the problem of the entire neural network getting stuck in local minima while enabling faster convergence. The error backpropagation method for complex-valued neural networks is as follows.

When using gradient descent for weight updates in complex-valued neural networks, Wirtinger calculus is employed for differentiating complex functions. As complex functions comprise real and imaginary parts, Wirtinger calculus conducts differentiation separately for the real and imaginary parts. For a complex number $z = x + jy$ defined with real and imaginary parts x and y , respectively, and a complex function $f(z) = u(x, y) + jv(x, y)$, the Wirtinger derivatives are defined as follows [14].

$$\frac{\partial f}{\partial z} = \frac{1}{2} \left(\frac{\partial f}{\partial x} - j \frac{\partial f}{\partial y} \right) = \frac{1}{2} \left(\frac{\partial u}{\partial x} + j \frac{\partial v}{\partial x} - j \frac{\partial u}{\partial y} + \frac{\partial v}{\partial y} \right) \quad (9)$$

$$\frac{\partial f}{\partial \bar{z}} = \frac{1}{2} \left(\frac{\partial f}{\partial x} + j \frac{\partial f}{\partial y} \right) = \frac{1}{2} \left(\frac{\partial u}{\partial x} + j \frac{\partial v}{\partial x} + j \frac{\partial u}{\partial y} - \frac{\partial v}{\partial y} \right) \quad (10)$$

$$df = \frac{\partial f}{\partial z} dz + \frac{\partial f}{\partial \bar{z}} d\bar{z} \quad (11)$$

In this context, $\bar{z}$ denotes the complex conjugate of z . In the Wirtinger derivative method, complex functions can be treated as extensions of real functions by dividing them into real and imaginary parts for differentiation. The following describes the updating process of complex variables using Step Size $\alpha/2$ and Loss Function L .

$$x_{n+1} = x_n - \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial x} \quad (12)$$

$$y_{n+1} = y_n - \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial y} \quad (13)$$

By substituting $z_{n+1} = x_{n+1} + j y_{n+1}$ into Equations (12) and (13) and then substituting the Wirtinger derivative at the end of the derivation process, it becomes as follows.

$$\begin{aligned} z_{n+1} &= x_n - \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial x} + j \left(y_n - \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial y} \right) \\ &= x_n - \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial x} + j y_n - j \left(\frac{\alpha}{2}\right) \frac{\partial L}{\partial y} \\ &= x_n + j y_n - \left(\frac{\alpha}{2}\right) \left(\frac{\partial L}{\partial x} + j \frac{\partial L}{\partial y} \right) \\ &= z_n - \alpha \frac{1}{2} \left(\frac{\partial L}{\partial x} + j \frac{\partial L}{\partial y} \right) \\ &= z_n - \alpha \frac{\partial L}{\partial \bar{z}} \end{aligned} \quad (14)$$

The partial derivative of L with respect to the complex conjugate $\frac{\partial L}{\partial \bar{z}}$ is referred to as the Conjugate Wirtinger Derivative. This demonstrates how the Conjugate Wirtinger Derivative can be utilized to simplify the update formula for complex variables. In terms of using the above method, rewriting Equation (5) as an output layer error yields Equation (15), where t_k represents the target value, and o_k represents the actual output value of output layer k .

$$e_k = t_k - o_k \quad (15)$$

And the Loss Function is defined as Equation (16).

$$L = e_k^2 = |t_k - o_k| \overline{|t_k - o_k|} \quad (16)$$

If the weights are large, the error is more likely to propagate back to the hidden layers. The hidden layer error e_j can be expressed by the following equation.

$$e_j = W_{jk}^T \cdot e_k \quad (17)$$

Using the result of Equation (14), the gradient of the error function with respect to the weight $\bar{W}_{ij}$ between input layer i and hidden layer j can be expressed as follows. Here, o_i is the output value of input layer i , and o_j is the output value of hidden layer j .

$$\frac{\partial L}{\partial \bar{W}_{ij}} = \frac{\partial L}{\partial o_j} \cdot \frac{\partial o_j}{\partial \bar{W}_{ij}} \quad (18)$$

$$\frac{\partial L}{\partial \bar{W}_{ij}} = -e_j \cdot \frac{\partial}{\partial \bar{W}_{ij}} P \left(\sum \bar{W}_{ij} \cdot \bar{o}_i \right) \quad (19)$$

$$\frac{\partial L}{\partial \bar{W}_{ij}} = -e_j \cdot P' \cdot \bar{o}_i \quad (20)$$

In Equations (19) and (20), P represents the phase-dependent activation function used in the hidden layer output, and P' is the derivative of the dependent activation function. The updating of hidden layer weights can be expressed as follows.

$$new\ W_{ij} = old\ W_{ij} - lr \cdot \frac{\partial L}{\partial \bar{W}_{ij}} \quad (21)$$

When dealing with complex variables, it is necessary to use $\frac{\partial L}{\partial \bar{W}_{ij}}$ instead of $\frac{\partial L}{\partial W_{ij}}$ for updating. Here, lr plays a role in adjusting the intensity of the change to prevent overshooting, and it is called the learning rate. Additionally, the derivative of the activation function P' is a complex function, and to find the derivative of a complex function, the Wirtinger derivative is used [15].

Complex-Valued Softmax: The softmax function, which is used as the activation function in the output layer, takes input values and normalizes each value to be between 0 and 1. It ensures that the sum of the normalized values is always 1. This function allows for the interpretation of results as probabilities. It can be expressed as Equation (22).

$$Softmax(Z)_i = \frac{e^{Z_i}}{\sum_{i=1}^n e^{Z_i}} \quad (22)$$

However, the softmax function can only interpret results as probabilities for real number inputs, and for complex number inputs, a different interpretation method is needed. In this paper, we propose a complex-valued softmax interpretation method for complex number inputs. The result of the softmax function for complex number inputs yields complex output values, and since the sum of all output values is $1 + 0j$, we propose an algorithm that calculates the distance between the output values and $1 + 0j$ in the complex plane and selects the index of the closest value. This can be expressed as Equation (24).

$$distance_i = |Softmax(Z)_i - \omega| \quad (23)$$

$$closest\_index = argmin(distance_i) \quad (24)$$

Here, $\omega = 1 + 0j$ is defined. Consequently, we decide to classify the complex number corresponding to $closest\_index$ as class 1.

5. Performance Analysis

In our experiments of speech signal recognition, a detailed comparison between the results of real-valued neural networks and complex-valued neural networks was conducted. Performance analysis was carried out alongside the pre-processing of complex-valued mel-frequency cepstral coefficients (MFCCs) for dimensionality reduction. Additionally, the impact of complex-valued MFCCs on speech recognition results was observed. Specifically, in recognizing given speech signals using neural networks, the influence of MFCC values and their complexification effect on model performance was evaluated and compared to determine whether it is possible to enhance performance while reducing the number of parameters.

Experimental Method: Figure 6 depicts the architecture of a multi-layer complex-valued neural network used for training and testing with a dataset consisting of 8000 short speech samples, each containing eight different words and lasting less than one second. The speech data used in this study are part of the speech command dataset collected by Google and distributed under the CC BY license.

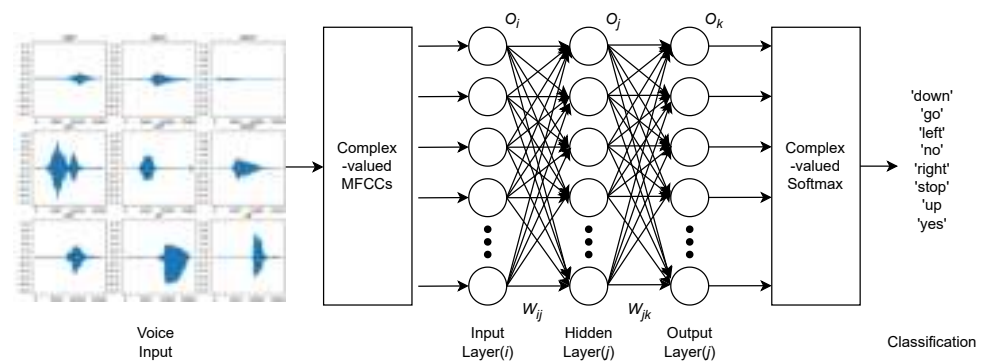


Figure 6. Experimental setup of multi-layer CVNN.

Complex-valued neural networks are structurally similar to real-valued neural networks, but the difference lies in the use of complex numbers, which introduces phase changes as the sum of products between inputs and weights rotates in the complex plane. These phase changes play an important role in complex-valued neural networks. The complex-valued neural network consists of an input layer with dimension-reduced complex-valued MFCCs obtained from the speech data through the complexification process shown in Figure 3, a hidden layer with 500 nodes, and an output layer with 8 nodes for classification. All nodes and weights use complex numbers. The final results were classified at the output layer. The dataset comprised 7000 samples for training and 1000 samples for testing. Eight-word audio clips were used as data, and experiments were conducted by dividing the input data based on whether complex-valued MFCC transformation was applied or not. In the case of using real-valued MFCCs, a real-valued neural network was employed, while a complex-valued neural network was used with complex-valued MFCCs. Both the real-valued and complex-valued neural networks were trained under the same conditions: $\text{lr} = 0.01$, batch size = 50, and number of epochs = 50. For the input data, the results of three different dimensionality reduction pre-processing steps were compared with the real-valued neural network. These pre-processing steps included MFCC dimensions of 20, 10, and 5. The length of the input signal for the neural network can be determined as shown in Equation (25). Specifically, the input speech signal was divided into 32 frames, and MFCCs were generated for each frame.

$$\text{The length of input signal data} = \text{MFCCs} \times \text{frames} \quad (25)$$

For instance, in the case of complex-valued MFCC 20, real-valued MFCC 40 is computed first. Using Equation (25), the length of the input signal for real-valued MFCC 40 becomes $40 \times 32 = 1280$. If we then convert these 40 MFCCs into complex numbers, as illustrated in Figure 3, dimensionality reduction is achieved to half the length, resulting in 16 frames. Consequently, the input signal length becomes $40 \times 16 = 640$, which is equivalent to the input signal length of real-valued MFCC 20. Similar dimensionality reduction can be performed for complex-valued MFCC 10 and complex-valued MFCC 5 using the same method. The total number of parameters for the entire neural network obtained through this process is presented in Table 1. In this table, $I(n)$ represents the number of neurons in the input layer, $H(n)$ represents the number of neurons in the hidden layer, and $O(n)$ represents the number of neurons in the output layer.

Table 1. Total number of parameters in neural network.

MFCC	Real-Valued MFCC	Complex-Valued MFCC
20	$I(20 \times 32 = 640) \times H(500) + H(500) \times O(8) = 324,000$	$I(40 \times 16 = 640) \times H(500) + H(500) \times O(8) = 324,000$
10	$I(10 \times 32 = 320) \times H(500) + H(500) \times O(8) = 164,000$	$I(20 \times 16 = 320) \times H(500) + H(500) \times O(8) = 164,000$
5	$I(5 \times 32 = 160) \times H(500) + H(500) \times O(8) = 84,000$	$I(10 \times 16 = 160) \times H(500) + H(500) \times O(8) = 84,000$

Complex-valued MFCC 20 vs. Real-valued MFCC 20: As shown in Figure 7, complex-valued MFCC 20 achieved a maximum training accuracy of 99.40% and a testing accuracy of 84.90% despite having the same input data length as real-valued MFCC 20, which achieved a maximum training accuracy of 98.13% and a testing accuracy of 83.10%. The results obtained using complex-valued MFCCs outperformed those obtained using real-valued MFCCs in terms of both training and testing accuracy.

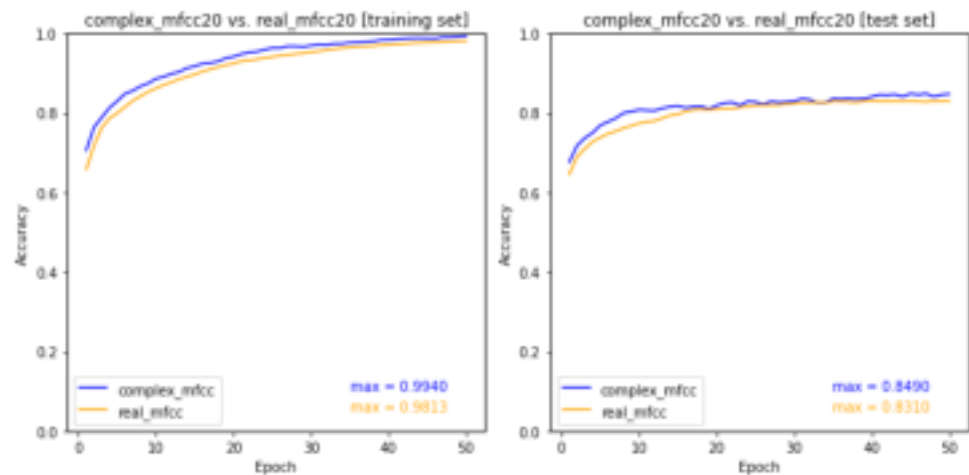


Figure 7. Comparison of complex-valued MFCC 20 and real-valued MFCC 20 results.

Complex-valued MFCC 10 vs. Real-valued MFCC 10: As shown in Figure 8, complex-valued MFCC 10 achieved a maximum training accuracy of 96.90% and a testing accuracy of 83.80% despite having the same input data length as real-valued MFCC 10, which achieved a maximum training accuracy of 94.24% and a testing accuracy of 81.20%. The results obtained using complex-valued MFCCs outperformed those obtained using real-valued MFCCs in terms of both training and testing accuracy.

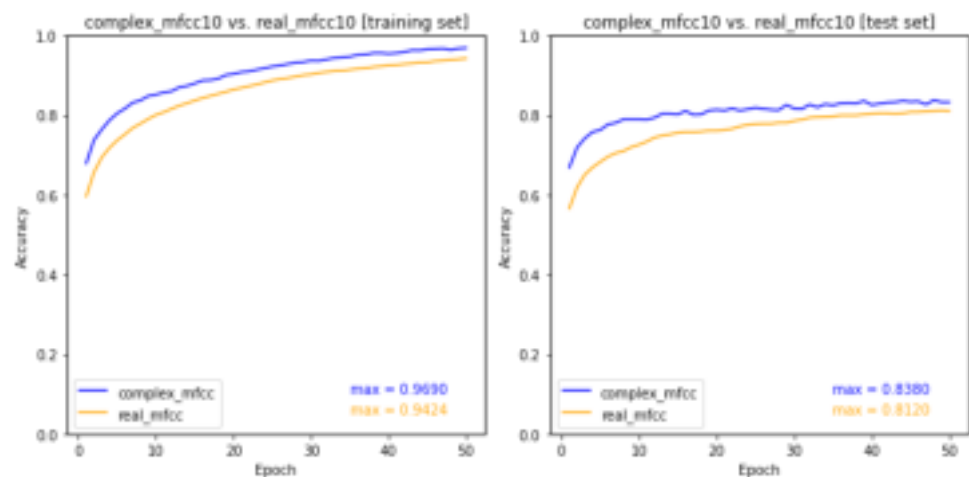


Figure 8. Comparison of complex-valued MFCC 10 and real-valued MFCC 10 results.

Complex-valued MFCC5 vs. Real-valued MFCC5: As shown in Figure 9, complex-valued MFCC 5 achieved a maximum training accuracy of 92.33% and a testing accuracy of 81.40% despite having the same input data length as real-valued MFCC 5, which achieved a maximum training accuracy of 86.34% and a testing accuracy of 74.90%. The results obtained using complex-valued MFCCs outperformed those obtained using real-valued MFCCs in terms of both training and testing accuracy.

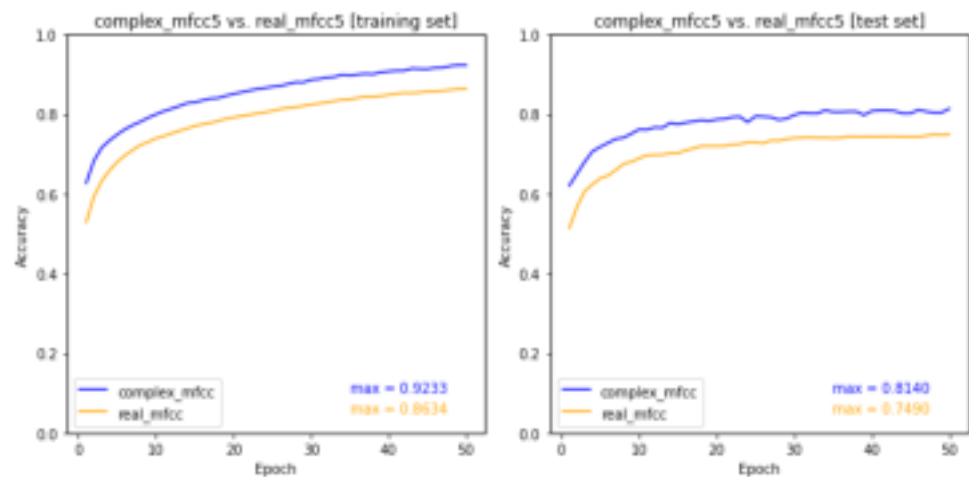


Figure 9. Comparison of complex-valued MFCC 5 and real-valued MFCC 5 results.

Overall Analysis Results: Upon reviewing the experimental results, it was observed that using complex-valued MFCCs yields higher performance compared to using real-valued MFCCs, even with the same input data length. Therefore, it was confirmed that employing complex-valued MFCCs and complex-valued neural networks leads to better results. The variation in accuracy according to the number of MFCC features is illustrated in Figure 10. Although not shown here, the same results were obtained using the F1 score.

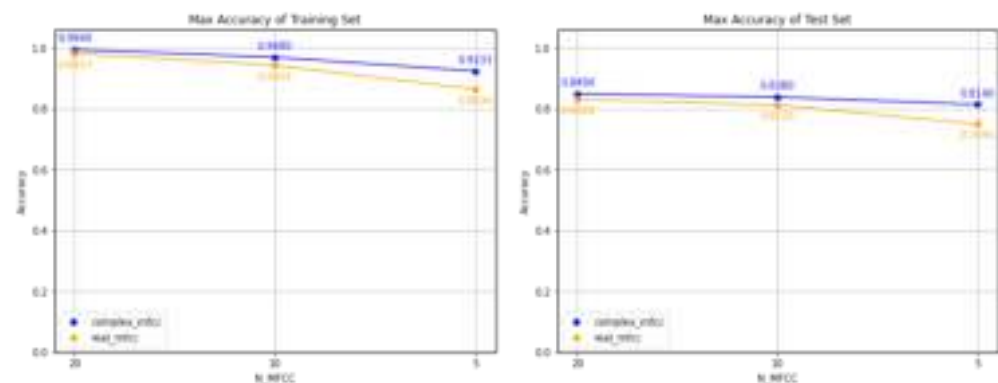


Figure 10. Accuracy variation with No. of MFCCs.

If there is a sufficient number of MFCCs, with increasing epochs, it appears that the performance using real numbers begins to converge towards that using complex numbers. However, Figure 9 shows a different outcome. Complex-valued MFCC 5 has 5 dimensionally reduced complex MFCCs obtained from 10 MFCCs through the complexification process shown in Figure 3. A real-valued MFCC 5 with only 5 MFCCs cannot catch up with complex-valued MFCC 5 even as the epochs increase. In other words, when the number of MFCCs is low, even as the epochs increase, the results of real-valued MFCCs do not match the performance of complex-valued MFCCs. As shown in Figure 10, despite having the same number of parameters, the results of complex-valued MFCC 5 are superior to those of real-valued MFCC 5. Therefore, it was observed that when using complex-valued MFCCs, the change in accuracy as the number of MFCC features decreases is smoother compared to when using real-valued MFCCs.

6. Conclusions

In this paper, we introduced complex-valued neural networks to address the problem of speech recognition. Our experiments utilized complex-valued MFCCs as inputs for a

complex-valued neural network. The results demonstrated superior performance when employing complex-valued MFCCs, with a more gradual change in accuracy as the number of MFCC features decreased, compared to using real-valued MFCCs. This suggests that complex-valued MFCCs can help to maintain performance with fewer features. These findings provide valuable insights into the potential of complex-valued neural networks and demonstrate the effectiveness of combining complex-valued MFCCs with complex-valued neural networks in speech recognition.

Despite these promising results, we acknowledge the certain limitations of our approach. This study was conducted on a limited dataset, which may not fully represent the diversity of real-world speech patterns. Moreover, the complex-valued neural network architecture described in this study is relatively simple and may not capture all the intricacies of speech signals.

Looking ahead, we aim to build upon this work to further enhance the accuracy and performance of speech signal processing and recognition models. Our future research will focus on designing more sophisticated complex-valued neural network models and conducting experiments on diverse datasets to improve performance. Additionally, we plan to explore the application of complex-valued neural networks to other fields beyond speech recognition, potentially expanding their use in various signal processing domains [16].

Author Contributions: Conceptualization, S.K.; project administration, M.P.; supervision, M.P.; writing—original draft, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was jointly supported by the National Research Foundation of Korea (NRF) via a grant provided by the Korea government (MSIT) (grant no. NRF-2023R1A2C1005461), and by the MSIT (Ministry of Science and ICT), Korea, under the Convergence Security Core Talent Training Business Support Program (IITP-2024-RS-2024-00426853) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation).

Data Availability Statement: The original contributions presented in the study are included in the article. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tebelskis, J. Speech Recognition Using Neural Networks. Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 1995.
2. Sarroff, A.M. Complex Neural Networks for Audio. Ph.D. Thesis, Dartmouth College, Hanover, NH, USA, 2018.
3. Abdul, Z.K.; Al-Talabani, A.K. Mel Frequency Cepstral Coefficient and Its Applications: A Review. *IEEE Access* **2022**, *10*, 122136–122158. [\[CrossRef\]](#)
4. Barrachina, J.A.; Ren, C.; Vieillard, G.; Morisseau, C.; Ovarlez, J.P. Theory and Implementation of Complex-Valued Neural Networks. *arXiv* **2023**, arXiv:2302.08286.
5. Aizenberg, I. *Complex-Valued Neural Networks with Multi-Valued Neurons*; Springer: Berlin/Heidelberg, Germany, 2016.
6. Graves, A.; Jaitly, N.; Mohamed, A.R. Hybrid speech recognition with Deep Bidirectional LSTM. In Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, 8–12 December 2013; pp. 273–278. [\[CrossRef\]](#)
7. Paz, P.; Garrido, M. Efficient Implementation of Complex Multipliers on FPGAs Using DSP Slices. *J. Signal Process. Syst.* **2023**, *95*, 543–550. [\[CrossRef\]](#)
8. Ahmad, M.; Zhang, L.; Chowdhury, M.E.H. FPGA Implementation of Complex-Valued Neural Network for Polar-Represented Image Classification. *Sensors* **2024**, *23*, 2627. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Anggraeni, D.; Sanjaya, W.S.; Nurasyidiek, M.Y.; Munawwaroh, M. The Implementation of Speech Recognition using Mel-Frequency Cepstrum Coefficients (MFCC) and Support Vector Machine (SVM) Method Based on Python to Control Robot Arm. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *288*, 012042. [\[CrossRef\]](#)
10. Dhanjal, A.S.; Singh, W. *A Comprehensive Survey on Automatic Speech Recognition Using Neural Networks*; Springer Nature: Berlin/Heidelberg, Germany, 2023.
11. Smith, S.W. *The Scientist and Engineer's Guide to Digital Signal Processing*, 2nd ed.; California Technical Publishing: San Diego, CA, USA, 1999; pp. 557–558.
12. Bassey, J.; Li, X.; Qian, L. A Survey of Complex-Valued Neural Networks. *arXiv* **2021**, arXiv:2101.12249v1.
13. MYONeuralNet. Complex-Valued Neural Networks—Experiments. 2016. Available online: <http://makeyourownneuralnetwork.blogspot.com/2016/05/complex-valued-neural-networks.html> (accessed on 19 February 2021).

14. Fischer, R. Wirtinger Calculus. In *Precoding and Signal Shaping for Digital Transmission*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2002; pp. 405–413.
15. PyTorch. Automatic Differentiation Package—torch.autograd, n.d. Available online: <https://pytorch.org/docs/stable/notes/autograd.html> (accessed on 3 November 2023).
16. Lee, C.; Hasegawa, H.; Gao, S. Complex-Valued Neural Networks: A Comprehensive Survey. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 1433–1454. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Job Recommendations: Benchmarking of Collaborative Filtering Methods for Classifieds

Robert Kwieciński ^{1,2} , Tomasz Górecki ^{2,*} , Agata Filipowska ³  and Viacheslav Dubrov ¹

¹ OLX Group, Królowej Jadwigi 43, 61-872 Poznań, Poland; r.kwiecinskipl@gmail.com (R.K.); slavadubrov@gmail.com (V.D.)

² Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Uniwersytetu Poznańskiego 4, 61-614 Poznań, Poland

³ Department of Information Systems, Poznań University of Economics and Business, Al. Niepodległości 10, 61-875 Poznań, Poland; agata.filipowska@ue.poznan.pl

* Correspondence: tomasz.gorecki@amu.edu.pl

Abstract: Classifieds pose numerous challenges for recommendation methods, including the temporary visibility of ads, the anonymity of most users, and the fact that typically only one user can consume an advertised item. In this work, we address these challenges by choosing models and evaluation procedures that are considered accurate, diverse, and efficient (in terms of memory and time consumption during training and prediction). This paper aims to benchmark various recommendation methods for job classifieds, using OLX Jobs as an example, to enhance the conversion rate of advertisements and user satisfaction. In our research, we implement scalable methods and represent different approaches to the recommendations: Alternating Least Square (ALS), LightFM, Prod2Vec, RP3Beta, and Sparse Linear Methods (SLIM). We conducted A/B tests by sending millions of messages with recommendations to perform online evaluations of selected methods. In addition, we have published the dataset created for our research. To the best of our knowledge, this is the first dataset of its kind. It contains 65,502,201 events performed on OLX Jobs by 3,295,942 users who interacted with (displayed, replied to, or bookmarked) 185,395 job ads over two weeks in 2020. We demonstrate that RP3Beta, SLIM, and ALS perform significantly better than Prod2Vec and LightFM when tested in a laboratory setting. Online A/B tests also show that sending messages with recommendations generated by the ALS and RP3Beta models increases the number of users contacting advertisers. Additionally, RP3Beta had a 20% more significant impact on this metric than ALS.

Keywords: job recommendations; classifieds; collaborative filtering; scalability testing; A/B tests; recommender systems; matrix factorization; graph-based recommender systems; OLX job interactions dataset; diversity metrics



Citation: Kwieciński, R.; Górecki, T.; Filipowska, A.; Dubrov, V. Job Recommendations: Benchmarking of Collaborative Filtering Methods for Classifieds. *Electronics* **2024**, *13*, 3049. <https://doi.org/10.3390/electronics13153049>

Academic Editors: Marco Lombardi, Domenico Santaniello, Zilu Liang and Muhammad Khan

Received: 30 May 2024

Revised: 24 July 2024

Accepted: 27 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Online classifieds are websites where advertisers post advertisements concerning the sale or rental of services or products [1]. This paper addresses online advertising using the example of OLX Jobs.

Classifieds strive to address the needs of both advertisers and responders to online ads. To match these groups, better recommendation methods are becoming more critical. Compared to traditional e-commerce (e.g., Amazon (<https://www.amazon.com/>, accessed on 26 June 2024) or eBay (<https://www.ebay.com/>, accessed on 26 June 2024)), classifieds must face different challenges, which we address in our evaluation procedure. One is the temporal visibility of ads, which we decided to address by frequently retraining the model. Hence, the model's efficiency, which we report, is essential to the evaluation. Another challenge is that an advertiser usually needs only one user to complete the transaction. This aspect was addressed by including diversity analysis in our assessment.

In our research, we focus on approaches emerging from collaborative filtering. Collaborative filtering models have proven their accuracy, even though they do not utilize additional information about users or items, which is sometimes hard to obtain or process in practice. This paper aims to evaluate different recommendation methods for use in the classifieds sector. The high sparsity of our dataset, along with the significant number of ads and users, contributes to the complexity of the research problem.

Below, we list our main contributions.

- We comprehensively evaluate diverse, scalable, and collaborative filtering approaches in a laboratory setting regarding accuracy, diversity, and efficiency. We evaluate the accuracy for different groups of users, depending on the number of items they interacted with, together with statistical analysis of differences. The proposed evaluation methodology addresses the major challenges of classifieds.
- We present the results of two online A/B tests demonstrating the impact of the ALS and RP3Beta models on OLX users. More than 1 million users participated in each test. The reported impact may be used to estimate the importance of recommender systems in the classifieds domain and enable other companies to make more informed decisions about developing such models.
- We publish and present a job interactions dataset. To our knowledge, this is the largest publicly available dataset on job interactions.

This paper consists of seven sections. Section 2 is a literature review on underlying developments. Section 3 describes the dataset developed and made publicly available on Kaggle. The design of the lab experiment, as well as the implementation of methods, are described in Section 4. Section 5 focuses on the results of the lab evaluation of our implementations as well as the statistical importance of the results. Section 6 presents the results of the online implementations performed during A/B tests with users, as well as a discussion of the results. Conclusions are presented in Section 7.

2. Related Work

In this section, we begin by outlining the unique characteristics of classifieds and job domains. We then provide an overview of recommender systems. Next, we discuss the factors that influence our selection of models and provide a detailed description of each model. Finally, we specify the research gaps.

2.1. Classified ad Sites

Classified ad sites (classifieds) are online versions of newspaper advertisement sections, with ads concerning vehicles, real estate, employment, pets, etc. Classifieds are a particular type of e-commerce site, as they do not support the finalization of transactions between parties—they are used instead to communicate about the product or service and to set up meetings while the transaction is carried out offline; this distinguishes them from traditional e-commerce sites such as eBay or Amazon [2].

Our research is focused on OLX, which has a presence in over 30 countries worldwide. One of the ad types published on classifieds is job offers. In our research, we identified several characteristics that distinguish job ads on classifieds from, for example, car or real estate ads:

- The requirements in an ad indicate the type of user who may respond to the ad—not all users possess the required competencies.
- More intensive involvement of the user during the search process, as finding a job is a primary need compared to, for example, buying a car.
- The job location dramatically impacts the user—to buy a mobile phone or a car, a user may travel or use a parcel service to deliver the item. Still, a job requires relocation if it is distant from the user's home.
- From a technical perspective, a job description has many qualitative aspects, and depending on the position, the differences may be significant even between ads from the same company.

On top of these features, during our research, we also identified the following challenges for recommendation methods on classifieds that influenced our implementation of the methods:

- The number of users and offers: with OLX Jobs, we dealt with millions of users and tens of thousands of offers, both constantly changing.
- A user does not have to create a profile to interact with ads, meaning that information from a profile cannot be used for recommendation, and recommendations are determined by user behavior during viewing.
- An ad usually has a limited number of visitors (interacting with the ad), as an ad usually concerns a unique offer. After the need is addressed, the ad is disabled.
- There is no information on whether a transaction occurred; on classifieds, conversion relates to receiving an answer to a published ad, as any transaction is offline.

These challenges create difficulties not only for implementation but also for testing the accuracy of the proposed methods. We considered these challenges while preparing the dataset and implementing and evaluating the methods described in this paper.

2.2. Recommendation Systems

It has been shown that the problem of overchoice [3] can decrease the consumption of products [4] and reduce revenue to the site [5]. Providers can ask their users for additional information about their expectations regarding items to reduce the number of choices presented. Even if the user is willing to spend time providing this information, the number of possibilities is usually too large to present to the user. To address this problem, many companies, such as Netflix [6] and Amazon [7], have created personalized recommendation systems whose role is to suggest relevant items to users [8].

Recommendations are used in multiple domains, including music, tourism, fashion, and food. Numerous studies have been made within the domain of e-commerce [9], classifieds [10], and job recommendations [11–13]. Li et al. [14] significantly improved their recommendations on LinkedIn by utilizing deep transfer learning to create domain-specific job-understanding models. Zhu et al. [15] proposed a cross-domain recommendation model utilizing a heterogeneous graph and community detection algorithm to recommend courses and jobs. Lacic et al. [16] evaluated different autoencoder architectures for session-based job recommendations and presented the results on popular job recommendation datasets.

Typically, we distinguish two approaches for providing recommendations: content-based recommendations and collaborative filtering [8].

In content-based recommendations [17], we utilize information about users (e.g., hobby, education, skills) and items (e.g., price, location, category). These systems usually use the interaction history (ratings, visits, purchases, and replies) of a single user but do not consider the ratings of all users simultaneously.

Collecting/processing information about users and items is not always feasible. In the collaborative filtering approach [18,19], we only utilize information about user interactions between items. Even without information on user features and item features, these systems can provide very accurate recommendations. We chose this approach because of the lack of user features and the difficulty of outperforming collaborative filtering by utilizing item features in a production environment (we elaborate on this in Section 6.3).

One of the most significant shortcomings of collaborative filtering is the impossibility of providing recommendations regarding users and items without interactions, known as the cold-start problem [20]. One possible solution to reducing this problem is the frequent retraining of the model. With many user interactions, this strategy requires a focus on the scalability of the model [21].

Another important factor determining the model choice is the type of interaction between a user and an item [20]. We call the feedback explicit when the user explicitly indicates their preference toward the item (e.g., rates it on a scale of 1–5). Another type is implicit feedback, when we can only assume the user's preference toward the item based

on user behavior (e.g., buying an item or watching a movie). Implicit feedback is less accurate, more abundant, and usually expresses only favorable preferences [22].

Laboratory evaluations of recommender systems are usually based on accuracy metrics [23]. Recently, much attention has been placed on other aspects, such as diversity, uncertainty, novelty, and coverage [24,25]. With classifieds, an item can usually be consumed by only one user, so we should avoid recommending the same item to a significant number of users. Hence, focusing on catalog coverage [26] is essential. One of the most challenging tasks in evaluating recommender systems is finding the offline metric most correlated with the online goal [27].

In this paper, we benchmark collaborative filtering models for a sizable implicit feedback dataset. We compare the accuracy, diversity, and efficiency of selected recommendation methods in a laboratory setting and then prove the methods' quality in online tests with users.

2.3. Recommendation Methods

Below, we list the most important aspects impacting method selection for our research.

- **Popularity in the literature.** We selected methods whose performances were thoroughly evaluated. All but one of the selected methods have been cited at least hundreds of times.
- **Popularity in industry.** Evaluating this aspect was challenging because companies rarely share the results of conducted experiments. Some popularity indicators include implementation availability and the number of people who use it (e.g., reflected in the number of stars and forks on GitHub). We also considered the experience of data scientists working within our company, who deployed most of the selected methods for some other use cases in the past.
- **Scalability.** We only considered solutions capable of handling millions of users and items. Additionally, we reduced the cold-start problem by frequently retraining the model. Hence, we had to choose techniques to be trained on our dataset within a few hours.
- **Diversity of approaches.** We wanted to test methods representing different "families" to assess their applicability to classifieds. We planned to identify the most promising family of approaches to continue research in this direction.

Finally, ALS, LightFM, Prod2Vec, RP3Beta, and SLIM were selected for testing and evaluation. The following subsections describe the reasons for choosing each specific method and the method definition. A common limitation of these methods is their inability to recommend items that are not visited by any user and generate recommendations for users who have not visited any item (cold-start problem, [21]). With our dataset, without user and item features, it is impossible to overcome this problem by any personalized method.

Selected methods have already been compared across different domains. Anelli et al. [28] compared RP3Beta, SLIM, ALS, and BPRMF (which is equivalent to LightFM when using the BPR loss function) on three datasets: Amazon Digital Music, Epinions, and MovieLens 1M, in terms of accuracy, diversity, and efficiency. The numbers of interactions in these datasets are, respectively, 145,523, 300,475, and 571,531. Our study utilized a dataset containing tens of millions of interactions, similar to those in OLX production systems. Additionally, we provide the results of an online evaluation.

Dacrema et al. [29] compared the accuracy of SLIM, RP3Beta, and ALS on several datasets (CiteULike-a, MovieLens 1M, Pinterest, Amazon Music, Amazon Movies, MovieLens 20M, Netflix, Yelp, and Gowalla). They showed that each of these algorithms outperformed the other two on at least one dataset. Some of their findings were further confirmed on MovieLens 1M and Pinterest datasets by Anelli et al. [30]. Zhu et al. [31] compared BPRMF, SLIM, and Prod2Vec on three datasets (Amazon Books, Yelp, and Gowalla), showing the superiority of SLIM. Dong et al. [32] analyzed the accuracy of RP3Beta, SLIM, and ALS on nine datasets, with SLIM providing the best results on eight datasets and ALS

on one. Unlike our study, these papers neither compared the selected methods in terms of diversity or efficiency nor conducted an online evaluation.

Recently, several recommendation methods based on neural networks have been proposed. It has been shown that adequately determined and tuned simple models often outperform them [28,31–33]. Additionally, due to computational complexity, models based on neural networks are less frequently tested on datasets containing tens of millions of interactions [28]. Hence, we leave it for future research to propose more advanced and scalable methods that provide superior results on our dataset.

2.3.1. Matrix Factorization Models

Matrix factorization techniques have been successfully utilized in recommender systems since the Netflix Prize competition [34]. The main idea is to approximate the sparse user–item interaction matrix as a product of two smaller, dense matrices. We evaluate two matrix factorization approaches: LightFM and ALS.

LightFM [35] model was proposed to overcome the cold-start problem of matrix factorization methods. In our case, it reduces to classical matrix factorization without additional information about users and items. Despite this, we decided to use this approach because of its efficient official implementation in supporting multiple loss functions: logistics, BPR [36], WARP [37], and k -OS WARP [38].

Let $\mathcal{U}$ be a set of users and $\mathcal{I}$ a set of items. For a given user $u \in \mathcal{U}$ and item $i \in \mathcal{I}$, we define $r_{ui} = 1$ if user u interacts with item i , and $r_{ui} = 0$, otherwise. The LightFM model predicts the score r_{ui} as follows:

$$\hat{r}_{ui} = \mathbf{x}_u \cdot \mathbf{y}_i + b_u + b_i,$$

where $\mathbf{x}_u$ and $\mathbf{y}_i$ denote user u 's and item i 's k -dimensional latent representations, and b_u and b_i are user u 's and item i 's biases. All of these parameters are learned during the training.

The logistic loss function should be used when both positive and negative feedback types are available, so it is inappropriate for our dataset, which we confirmed experimentally during the hyperparameter tuning.

The BPR, WARP, and k -OS WARP loss functions are pairwise learning-to-rank approaches that are more relevant for our top- k recommendations task. They try to modify model parameters to maximize the difference between the scores of the items $i \in \mathcal{D}_u$ with which user u interacts and of the items $i \notin \mathcal{D}_u$ with which the user does not interact with.

Specifically, in the BPR loss function [36], we attempt to maximize the expression, as follows:

$$\sum_{(u,i,j) \in \mathcal{S}} \ln \sigma(\hat{r}_{ui} - \hat{r}_{uj}) - \lambda_{\Theta} \|\Theta\|^2,$$

where

$$\mathcal{S} = \{(u, i, j) | u \in \mathcal{U}, i \in \mathcal{D}_u \text{ and } j \notin \mathcal{D}_u\},$$

$$\sigma(x) = \frac{1}{1 + e^{-x}},$$

λ_{Θ} is a model-specific regularization parameter, Θ is a parameter vector, and $\|\cdot\|$ is the Euclidean norm.

We present WARP [37] as a case of k -OS WARP [38]. The k -OS WARP loss function is a sum of losses for each user:

$$\sum_{u \in \mathcal{U}} L(\hat{\mathbf{r}}_u, \mathcal{D}_u),$$

where $\hat{\mathbf{r}}_u$ is a vector of estimated scores $\hat{r}_{ui}$ for each item $i \in \mathcal{I}$.

For a given user u , we order the interacted items $\mathcal{D}_u$ in decreasing order of the estimated scores. Namely, we order the items as $i_1, i_2, \dots, i_{|\mathcal{D}_u|}$, where the indices are chosen in such a way that we have the following:

$$\hat{r}_{ui_1} \geq \hat{r}_{ui_2} \geq \dots \geq \hat{r}_{ui_{|\mathcal{D}_u|}}.$$

Then, we have the following:

$$L(\hat{\mathbf{r}}_u, \mathcal{D}_u) = \frac{1}{Z} \sum_{s=1}^{|\mathcal{D}_u|} P(s) \Phi(\text{rank}_{i_s}(\hat{\mathbf{r}}_u)),$$

where $\text{rank}_i(\hat{\mathbf{r}}_u) = \sum_{j \in I \setminus \mathcal{D}_u} \mathbb{1}(1 + \hat{r}_{uj} \geq \hat{r}_{ui})$, $\mathbb{1}$ is an indicator function, $\Phi(n) = \sum_{m=1}^n \frac{1}{m}$, and $Z = \sum_{s=1}^{|\mathcal{D}_u|} P(s)$ normalizes the weights induced by P .

In practice, we sample a positive item i concerning the weighting function P and randomly pick unseen items j until, after N samplings, $1 + \hat{r}_{uj} \geq \hat{r}_{ui}$. Then, we perform a gradient step to minimize $\Phi(\lfloor \frac{|I \setminus \mathcal{D}_u|}{N} \rfloor)(1 - \hat{r}_{ui} + \hat{r}_{uj})$.

Finally, the k -OS WARP loss is achieved for $P(k) = 1$ and $P(m) = 0$ if $m \neq k$. The WARP loss is achieved when we set $P(m) = 1$ for $m \in \mathbb{N}$. We can skip ordering positive items from $\mathcal{D}_u$ to simplify the computations.

In our research, all loss functions (logistic, BPR, WARP, and k -OS WARP) were treated as hyperparameters of the LightFM model and optimized during training.

The ALS model (alternating least squares), also known as WRMF—weighted regularized matrix factorization [39], is an example of a matrix factorization method designed for implicit feedback datasets. We chose this method because of its proven performance, scalability, and popularity in research and industry. Additionally, the ALS model has already been implemented and used in OLX.

We define the confidence of an observation as $c_{ui} = 1 + \alpha r_{ui}$, where α is a hyperparameter. We also define the preference as $p_{ui} = \mathbb{1}(r_{ui} > 0)$. Then, in ALS, we optimize the following expression:

$$\min_{\mathbf{x}_*, \mathbf{y}_*} \sum_{u,i} c_{ui} (p_{ui} - \mathbf{x}_u^\top \mathbf{y}_i)^2 + \lambda (\sum_u \|\mathbf{x}_u\|^2 + \sum_i \|\mathbf{y}_i\|^2),$$

where $\mathbf{x}_u$ is user u 's f -dimensional embedding, $\mathbf{y}_i$ is item i 's f -dimensional embedding, and λ is a regularization parameter. The model's name refers to the optimized method for this function. The alternating least squares procedure assumes that user or item latent factors are fixed and analytically determine the others. The procedure is efficiently repeated a few times [39,40].

In matrix factorization approaches, the embedding for each user is learned separately, which might result in high time and memory utilization if the number of users is vast. Additionally, the recommendations might lack explainability because two users with the same interactions might receive different recommendations. Other considered models (SLIM, RP3Beta, Prod2Vec) provide the same set of recommendations for users with the same interactions.

2.3.2. Neighborhood-Based Models

The SLIM model (sparse linear methods for Top-N recommender systems [41]) is based on item-based nearest neighbors regression [8]. We chose this method because of its advantages over item-based k -nearest-neighbors and matrix factorization models [41] and good performance against non-neural and neural approaches [29].

In this model, we approximate the user–item matrix $\mathbf{R}$ by learning the item–item sparse similarity matrix $\mathbf{W}$. Explicitly, $\mathbf{W}$ is learned by minimizing the following expression:

$$\frac{1}{2} \|\mathbf{R} - \mathbf{R}\mathbf{W}\|_F^2 + \frac{\beta}{2} \|\mathbf{W}\|_F^2 + \lambda \|\mathbf{W}\|_1,$$

where $w_{ij} \geq 0$ and $w_{ii} = 0$ for all $i, j, 1 \leq i, j \leq |I|$, $\|\cdot\|_F$ is the matrix Frobenius norm:

$$\|\mathbf{W}\|_F = \sqrt{\sum_{i=1}^{|I|} \sum_{j=1}^{|I|} w_{ij}^2},$$

$\|\cdot\|_1$ is the entry-wise ℓ_1 -norm:

$$\|\mathbf{W}\|_1 = \sum_{i=1}^{|I|} \sum_{j=1}^{|I|} |w_{ij}|$$

and β and λ are regularization parameters.

We can see that each column of $\mathbf{W}$ can be learned independently, which improves the scalability of this model [41].

A limitation of this method is that it does not generate user and item embeddings, which could be helpful for other tasks (e.g., clustering of users or items). We can treat the columns of the matrix $\mathbf{W}$ as item embeddings, but they are sparse $|I|$ -dimensional vectors, so they probably cannot be directly utilized in other tasks.

2.3.3. Graph-Based Approaches

RP3Beta [42] is a graph-based collaborative filtering approach. Similarly to Dacrema et al. [29], we present this model as an extension of the P3alpha model [43]. We consider a bipartite and undirected graph $G = (\mathcal{V}, \mathcal{E})$, where each vertex $v \in \mathcal{V}$ represents either a user or item and each edge $\{v_u, v_i\} \in \mathcal{E}$ represents the interaction between a given user u and a given item i . Let $\mathbf{A}$ be the adjacency matrix and $\mathbf{D}$ the degree matrix of the graph G . Then, the scores of P3alpha are stored in the matrix

$$((\mathbf{D}^{-1}\mathbf{A})^{\circ\alpha})^3,$$

where $\circ$ denotes the element-wise power of a matrix (Hadamard power). More precisely, the score $\hat{r}_{ui}$ is stored on the intersection of the row representing user u and the column representing item i . We can see that this score is a sum of scores assigned to paths of length 3 connecting user u and item i in the graph G .

In RP3Beta, the score is computed with P3alpha and divided by the popularity of the items raised to the power of β to mitigate the popularity bias.

We chose this model because of its simplicity, scalability, and proven performance [29,30], especially given the high sparsity of our dataset, which significantly decreases memory utilization when training the model. This enables us to directly compute and store the item-item similarity matrix in memory instead of the random walks approximation proposed in the original paper.

RP3Beta, similar to SLIM, does not generate user and item embeddings. Another related limitation is its inability to recommend items whose distance on the user-item bipartite graph is greater than three from a given user. Specifically, item i can be recommended to user u only if at least one user u' has visited item i and at least one of the items visited by user u . Additionally, the RP3Beta model does not have parameters optimized during training; hence, it might be unable to exploit the complex relations in data.

2.3.4. Prod2Vec

The Prod2Vec [44] (or Item2vec [45]) model is based on the Word2Vec model [46], widely used in natural language processing. The model focuses on learning vector representations of items using a sequence of items as a “sentence” and items within that sequence as “words”. This method was chosen because it is scalable and very different from the compared methods. Additionally, this approach is based on the current state-of-the-art hybrid deep neural network recommender system used in other categories at OLX.

Prod2Vec uses the skip-gram model by maximizing the objective function over the set S of item sequences, defined as follows:

$$L = \sum_{s \in S} \sum_{i_k \in s} \sum_{\substack{-c \leq l \leq c \\ l \neq 0}} \log \mathbb{P}(i_{k+l}|i_k),$$

where c is the context length for item sequences, and items from the same sequence are ordered arbitrarily. The probability $\mathbb{P}(i_{k+l}|i_k)$ of observing a neighboring item i_{k+l} given the current item i_k is defined using the soft-max function:

$$\mathbb{P}(i_{k+l}|i_k) = \frac{\exp(\mathbf{y}_{i_k}^\top \mathbf{y}'_{i_{k+l}})}{\sum_{i=1}^{|I|} \exp(\mathbf{y}_{i_k}^\top \mathbf{y}'_i)},$$

where $\mathbf{y}_i$ and $\mathbf{y}'_i$ are the input and output vector representations of item i , and $|I|$ is the number of unique items in the vocabulary. Prod2Vec models the context of an item sequence, where items with similar contexts (i.e., with similar neighboring interactions) have similar vector representations.

Before generating recommendations, we normalize all input vector representations, which improves efficiency. To generate recommendations, we represent the user as an average of input vector representations of interacted items. Then, we search for items whose input representations have the highest cosine similarity with the user representation (k -nearest neighbors in the latent space).

A potential limitation of this method is a strict dependency between user and item representations. Some interactions should potentially have a greater impact on user representations than others (which is possible in matrix factorization models).

2.4. Research Gap

This paper aims to benchmark different recommendation methods in the job classifieds domain, addressing a specific research gap. To the best of our knowledge, the recommendation methods addressed have never been tested or benchmarked on a large scale while working online in the job domain. In this work, we compare these methods in a laboratory setting using the dataset we published and in online A/B tests carried out in a production environment.

3. OLX Job Interactions Dataset

This research provides a commercial dataset that contains 65,502,201 events performed on OLX Jobs by 3,295,942 users who interacted with 185,395 job ads in two weeks in 2020. We published the dataset on Kaggle (<https://www.kaggle.com/olxdatascience/olx-jobs-interactions>, accessed on 26 June 2024).

Each row of the dataset represents the event a given user performs regarding a given ad at a given time (Table 1). The following events involving user behaviors were included:

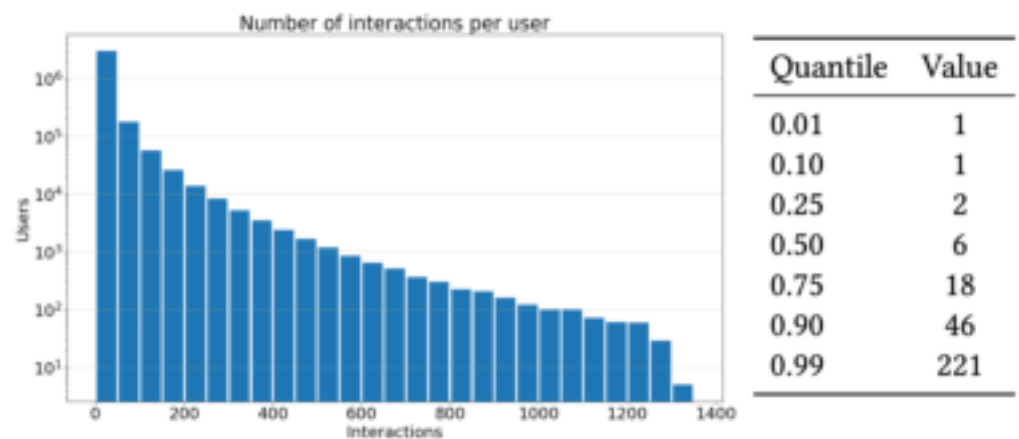
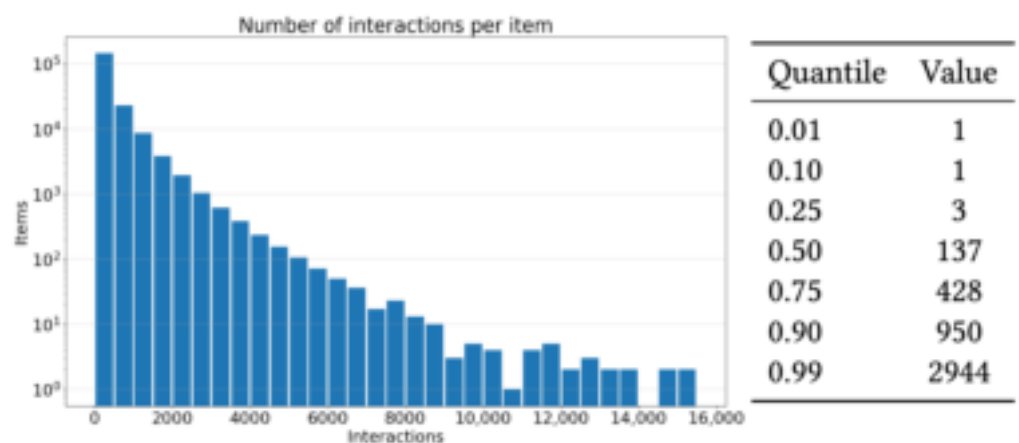
- click: the user visited the item detail page;
- bookmark: the user added the item to bookmarks;
- chat_click: the user opened the chat to contact the item's owner;
- contact_phone_click_1: the user revealed the phone number attached to the item;
- contact_phone_click_2: the user clicked to make a phone call to the item's owner;
- contact_phone_click_3: the user clicked to send an SMS message to the item's owner;
- contact_partner_click: the user clicked to access the item's owner's external page;
- contact_chat: the user sent the item's owner a message.

Table 1. Example rows from the dataset.

User	Item	Event	Timestamp
1745587	168661	click	1582216025
843008	62838	click	1582485868
14285	30469	bookmark	1582247367
1142944	80122	click	1581805847
2835659	23728	chat_click	1582397836

Source: own study.

The data sparsity (% of missing entries in the user–item matrix) equals 99.9923%, which is very low but typical for classifieds (for comparison, the data sparsity on the MovieLens 20M Dataset is 99.47%, making it almost 70 times denser). Distributions of interactions per user, interactions per item, and events are provided in Figures 1 and 2, and Table 2, respectively. We used a logarithmic scale because the distributions are strongly right-skewed.

**Figure 1.** Number of interactions per user. Source: own study.**Figure 2.** Number of interactions per item. Source: own study.

According to De Ruijt and Bhulai [47], the most popular datasets used for the evaluation of job recommendations systems are CareerBuilder12 (<https://www.kaggle.com/competitions/job-recommendation>, accessed on 16 June 2024), RecSys16 [12], and RecSys17 [13]. Table 3 shows the statistics of considered datasets. The number of unique interactions is the number of unique user–item pairs where the user interacted with the item. Even though researchers use the RecSys16 and the RecSys17 datasets, they were published during data science competitions under strictly limited conditions and are no longer available. Hence, we could not calculate the number of unique interactions for these

datasets. We did observe that the OLX job interactions dataset has the highest number of interactions and types among the considered datasets.

Table 2. Distribution of events.

Action	Frequency (%)
click	89.794
contact_phone_click_1	2.628
bookmark	2.511
chat_click	2.136
contact_chat	1.448
contact_partner_click	0.701
contact_phone_click_2	0.679
contact_phone_click_3	0.103

Source: own study.

Table 3. Comparison between job recommendation datasets. The letter *D* denotes the standard deviation.

	OLX Jobs Interactions	CareerBuilder12	RecSys16	RecSys17
Interactions	65.50 M	1.60 M	8.83 M	8.27 M
Unique interactions	47.17 M	1.60 M	-	-
Users	3.30 M	0.3 2M	1.37 M	1.50 M
Average number of unique interactions per user	14.31 (<i>D</i> = 29.23)	4.99 (<i>D</i> = 11.42)	<6.46	<5.51
Items	0.19 M	0.37 M	1.36 M	1.31 M
Average number of unique interactions per item	254.42 (<i>D</i> = 426.02)	4.38 (<i>D</i> = 8.19)	<6.50	<6.31
Density of the interaction matrix (%)	0.0077%	0.0014%	<0.0005%	<0.0004%
Types of interactions	8	1	4	5
Timestamps	✓	✓	✓	✓
User features	✗	✓	✓	✓
Item features	✗	✓	✓	✓

Source: own study.

It should be noted that maintaining the confidentiality of ads and users was a priority when preparing this dataset. The measures taken to protect privacy included the following:

- Original user and item identifiers were replaced with unique random integers;
- Some undisclosed constant integer was added to each timestamp;
- Some fractions of interactions were filtered out;
- Some additional artificial interactions were added.

4. Experimental Setup

4.1. Implementation

We implemented the methods mentioned above based on previous developments. Details on how our implementations differ from the previously tested approaches are provided in Table 4. The implementation of the methods for reproducibility of the results is available on GitHub (<https://github.com/rob-kwiec/olx-jobs-recommendations>, accessed on 29 May 2024).

Table 4. Methods re-implemented for the research.

Method	Family of Methods	Source	Difference from the Original Method
LightFM	Matrix factorization	Implementation based on: [48]. Supporting paper: [35].	None
ALS	Matrix factorization	Implementation based on: [49]. Supporting papers: [39,40].	None
SLIM	Neighborhood-based	Sources that inspired our implementation: [50,51]. Supporting paper: [41].	None
RP3Beta	Graph-based	Source that inspired our implementation: [52]. Supporting papers: [29,53].	Performed direct computations on sparse matrices instead of random walks approximation.
Prod2Vec	Word2Vec	Implementation based on: [54]. Supporting papers: [44,45].	Sequences of interactions, ordered by a timestamp. Representing the user as an average of the representations of interacted items. We also experimented with CBOW (continuous bag of words).

Source: own study.

4.2. Laboratory Research and Online Testing Goals

Our ultimate business goal was to increase the number of users applying for jobs through OLX Jobs. To achieve that, we try to solve a ranking problem; more precisely, we recommend ten items with the highest chance of receiving a given user's interaction. The issue of finding an offline metric most correlated with the application rate in the job domain was considered by Mogenet et al. [27]. Since our goal is similar, we followed their results by optimizing for precision@10 (precision for the first ten recommended items [23]). However, we also report and discuss other accuracy, diversity, and efficiency metrics.

We are focused on users actively looking for jobs by interacting with job ads. Therefore, we do not consider users without interactions (cold-start users).

4.3. Data Split

The dataset described in Section 3 was split into training-validation and test datasets as required by the research methodology. The test set includes 20% of the newest interactions. This means that out of 14 days, approximately 3 days were included in the test set (see Figure 3). This way of splitting the dataset into training and test sets is the most realistic setting according to Meng et al. [55], who studied several splitting strategies for evaluating recommendation systems.

In both datasets, we also limited the number of interactions to unique interactions between a user and an item. If a user interacted more than once with an item, only the first interaction was counted, and the timestamp of that interaction was associated with the item. We also did not distinguish between types of interactions. In addition, two more modifications were applied: only users who appeared in the training and validation dataset were included in the test set (because we could not provide personalized recommendations to other users without additional knowledge about them), and we filtered out all user-item pairs that were present in the training set (to avoid recommending items that had already been seen). As a result, we ended up with about 38 M rows in the training and validation set and about 6 M rows in the test set. More statistics of these two datasets are presented in Table 5.

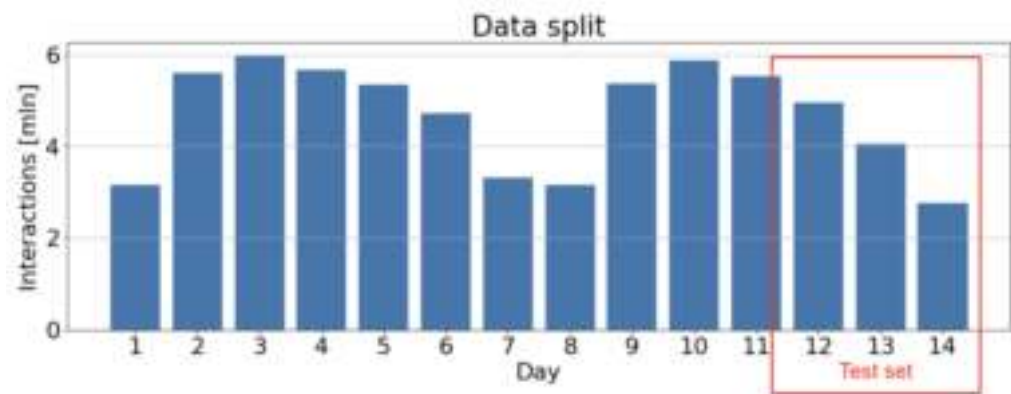


Figure 3. Dataset preparation (split into training–validation and testing, depending on the number of days). Source: own study.

Table 5. Comparison between training/validation and test sets. The letter *D* denotes standard deviation.

	OLX Job Interactions	Training and Validation Set	Test Set
Interactions	65.50 M	52.40 M	6.11 M
Unique interactions	47.17 M	38.25 M	6.11 M
Users	3.30 M	2.83 M	0.62 M
Average number of unique interactions per user	14.31 (<i>D</i> = 29.23)	13.50 (<i>D</i> = 26.11)	9.87 (<i>D</i> = 14.38)
Items	0.19 M	0.18 M	0.13 M
Average number of unique interactions per item	254.42 (<i>D</i> = 426.02)	214.05 (<i>D</i> = 365.93)	47.51 (<i>D</i> = 83.84)
Density of the interaction matrix (%)	0.0077%	0.0076%	0.0077%

Source: own study.

4.4. Model Tuning

For the sake of cost and time, when performing the optimization of parameters, we made a restriction to 20% of users and 20% of items of the training and validation set. Then, we split the restricted dataset into the training–validation dataset with the rules described in the section above. For each model, there were 100 iterations chosen by Bayesian optimization using Gaussian processes implemented in scikit-learn (https://scikit-optimize.github.io/stable/modules/generated/skopt.gp_minimize.html, accessed on 29 May 2024). The configuration file defines spaces for hyperparameters and is available in our code repository (<https://github.com/rob-kwiec/olx-jobs-recommendations/blob/main/src/tuning/config.py>, accessed on 29 May 2024). For example, for RP3Beta, we restricted ourselves to searching for α and β in the interval $[0, 2]$.

Recommendations in the optimization process were prepared for 30,000 users, and as mentioned above, the optimization score used was precision@10. Figure 4 shows the precision score depending on the parameters. It is visible that ALS was not highly optimized within the process, but for other methods, this optimization was crucial to improving precision. Hyperparameters for all models are presented in Table 6.

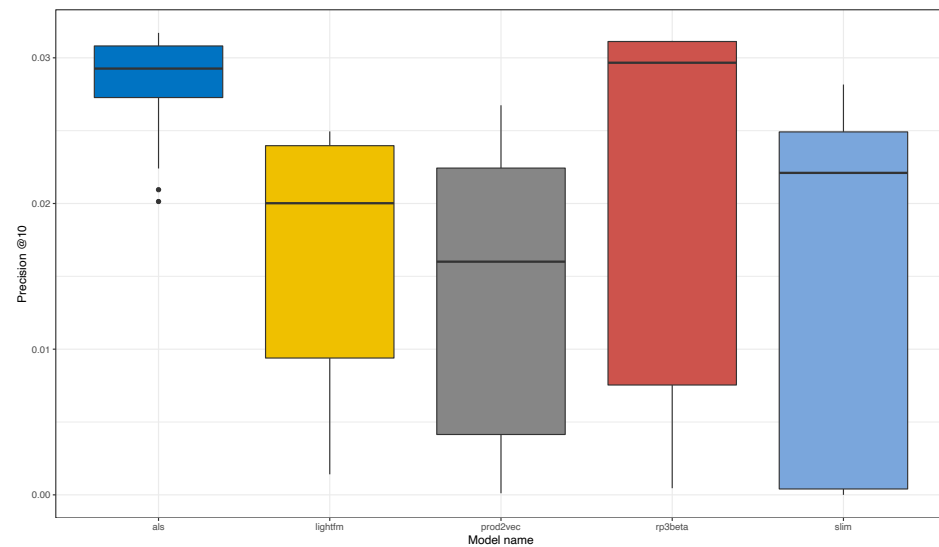


Figure 4. Precision for each model, depending on the parameters. Source: own study.

Table 6. Model hyperparameters.

Model	Model Hyperparameters
als	{‘factors’: 357, ‘regularization’: 0.001, ‘iterations’: 20, ‘event_weights_multiplier’: 63}
lightfm	{‘no_components’: 512, ‘k’: 3, ‘n’: 20, ‘learning_schedule’: ‘adadelta’, ‘loss’: ‘warp’, ‘max_sampled’: 61, ‘epochs’: 11}
prod2vec	{‘vector_size’: 168, ‘alpha’: 0.028728, ‘window’: 20, ‘min_count’: 16, ‘sample’: 0.002690026, ‘min_alpha’: 0.0, ‘sg’: 1, ‘hs’: 1, ‘negative’: 200, ‘ns_exponent’: −0.16447846705441527, ‘cbow_mean’: 0, ‘epochs’: 22}
RP3Beta	{‘alpha’: 0.61447198, ‘beta’: 0.1443548}
slim	{‘alpha’: 0.00181289, ‘l1_ratio’: 0.0, ‘iterations’: 3}

Source: own study.

4.5. Model Evaluation

We evaluate our models in terms of accuracy, diversity, and efficiency.

The accuracy of the recommendation method on implicit feedback datasets is typically evaluated by examining which items from the list of the top k recommendations the user interacted with in the test set [8]. Many evaluation metrics determine the metrics for each user separately and present the mean as the final evaluation score of the method [23]. The recommendation system may be considered a classifier, where the user’s interactions in the test set are positive examples, and the top k recommendations are items classified as positive by the model. Therefore, methods used for evaluating classifiers can be directly applied to recommender systems. For instance, the value of $\text{precision}@k$ for a given user is the number of recommended items the user interacted with in the test set divided by the number k of items recommended to that user. Other metrics discussed in this work were described by Tamm et al. [23].

In the classifieds domain, primarily the jobs domain, only one user is usually needed to complete a transaction regarding a given ad. In this case, the diversity of recommendations is even more critical than in other domains (like music, movies, or other e-commerce, excluding classifieds). To assess this aspect, we use three measures: test coverage, Shannon entropy [56], and the Gini index [56]. Test coverage refers to the fraction of test items recommended to at least one user. With Shannon entropy and the Gini Index, we ignore items outside the test set. Greater test coverage and Shannon entropy values indicate higher diversity, whereas greater values of the Gini index indicate lower diversity.

The final laboratory evaluation step focuses on efficiency. For this purpose, we utilized an AWS SageMaker ml.m5.4xlarge instance equipped with 64 GB of RAM and 16 vCPUs, featuring an Intel(R) Xeon(R) Platinum 8259CL CPU @ 2.50GHz, running on Amazon Linux AMI 2018.03. All methods were evaluated concerning three functions: data preprocessing, model fitting, and item recommendation. The execution time was calculated for the total time needed to compute the model (in hours). The memory peak was calculated in GB, and maximum memory usage was considered when executing a given function. This evaluation was performed using the tracemalloc library (<https://docs.python.org/3/library/tracemalloc.html>, accessed on 29 May 2024).

5. Laboratory Evaluation

This section describes the laboratory evaluation of the methods discussed in the developed dataset. We used the best-performing hyperparameters found in the previous section. Each model was trained on the entire training and validation dataset, and recommendations were produced for all 619,389 users from the test set. To better present the values of each offline metric, we compared the described methods with simple, non-personalized baselines: the most popular product approach and random recommendations.

5.1. Accuracy

All metrics in this section consider the first ten recommendations, e.g., precision@10, recall@10. Higher metric values indicate a higher quality of recommendations. The values should not be directly compared with the results achieved on other datasets because metrics heavily depend on the distribution of the dataset (for example, high sparsity) and the train/test splitting strategy [55].

5.1.1. Evaluation of Accuracy

Table 7 presents the outcomes of the evaluation. Most of the observed differences are statistically significant due to the large number of evaluated users. We elaborate more on the statistical significance of the results in Section 5.1.2.

Table 7. Accuracy: evaluation results. All presented metrics were described by Tamm et al. [23]. For the mAP metric, we used the variant with $x = \min(k, r)$.

Metric	RP3Beta	SLIM	ALS	Prod2Vec	LightFM	Most Popular	Random
precision	0.0484	0.0472	0.0434	0.0368	0.0359	0.0012	0.00006
recall	0.0783	0.0736	0.0657	0.0580	0.0564	0.0012	0.00005
ndcg	0.0759	0.0721	0.0657	0.0567	0.0545	0.0016	0.00007
mAP	0.0393	0.0365	0.0329	0.0282	0.0264	0.0006	0.00002
MRR	0.1365	0.1314	0.1230	0.1065	0.1034	0.0038	0.00019
LAUC	0.5391	0.5368	0.5328	0.5289	0.5281	0.5006	0.49999
HR	0.3131	0.3066	0.2878	0.2537	0.2547	0.0112	0.00059

Source: own study.

Our approaches significantly outperform random and most popular item recommendations. The low performance of the most popular approach results from the nature of classifieds, primarily the domain of the job, where users are mostly interested in a specific location and category. Building more sophisticated, personalized recommendation systems is reasonable in such a scenario.

It is demonstrated that RP3Beta outperforms other approaches in terms of accuracy metrics. We believe that this is related to the high sparsity of our dataset. RP3Beta calculates the recommendations deterministically (by leveraging paths of length three on a user-item bipartite graph). All other approaches utilize machine learning techniques to find user representations (ALS, LightFM), item representations (ALS, LightFM, Prod2Vec), or direct item-item similarities (SLIM), which might be in cases where users or items have a low number of interactions.

5.1.2. Statistical Comparison for Precision@10

We present a detailed statistical comparison for our main metric, precision@10, to identify differences between the methods. To begin with, we test the null hypothesis that all methods perform the same and the observed differences are merely random (omnibus test). The Friedman test [57,58] with the Iman and Davenport [59] extension is probably the most popular omnibus test, and it is usually a good choice when comparing more than five different algorithms [60,61]. Let R_{ij} be the rank of the j th of K methods on the i th of N samples and

$$R_j = \frac{1}{N} \sum_{i=1}^N R_{ij}.$$

In our case, the number of samples, N , is the number of users in the test set, $N = 619,389$. The test compares the mean ranks of methods and is based on the following statistics:

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1) - \chi_F^2},$$

where

$$\chi_F^2 = \frac{12N}{K(K+1)} \sum_{i=1}^K R_i^2 - 3N(K+1)$$

is the Friedman statistic, which has the F distribution with $K-1$ and $(K-1)(N-1)$ degrees of freedom. The p -value from this test is equal to 0. The obtained p -value indicates that we can safely reject the null hypothesis that all algorithms perform the same. Therefore, we can proceed with the post hoc tests to detect significant pairwise differences among all the methods. Demšar [62] proposes the use of the Nemenyi test [63], which compares all algorithms pairwise. For a significance level α , the test determines the critical difference (CD). If the difference between the average rankings of two algorithms is greater than

$$CD = q_\alpha \sqrt{\frac{K(K+1)}{6N}},$$

the null hypothesis that the algorithms have the same performance is rejected (q_α is based on the studentized range statistic divided by $\sqrt{2}$). Demšar [62] proposed a plot to check the differences (the CD plot visually). Algorithms not joined by a line can be considered different in the plot.

In our case, with a significance of $\alpha = 0.05$, any two algorithms with a difference in the mean rank above 0.0114 will be regarded as non-equal (Figure 5).

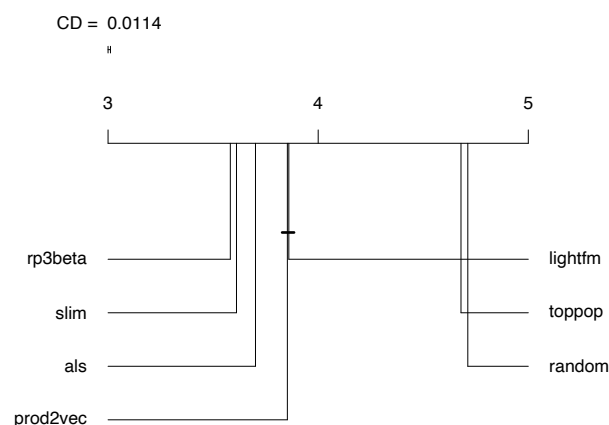


Figure 5. Critical difference plot (for precision). Source: own study.

5.1.3. Accuracy for Users with Different Numbers of Interactions

The only information we have about our users is their interactions. In Figure 6, we examine how the number of items a user has interacted with influences our main metric, precision@10. We divide our users into ten similarly sized groups based on the number of interactions.

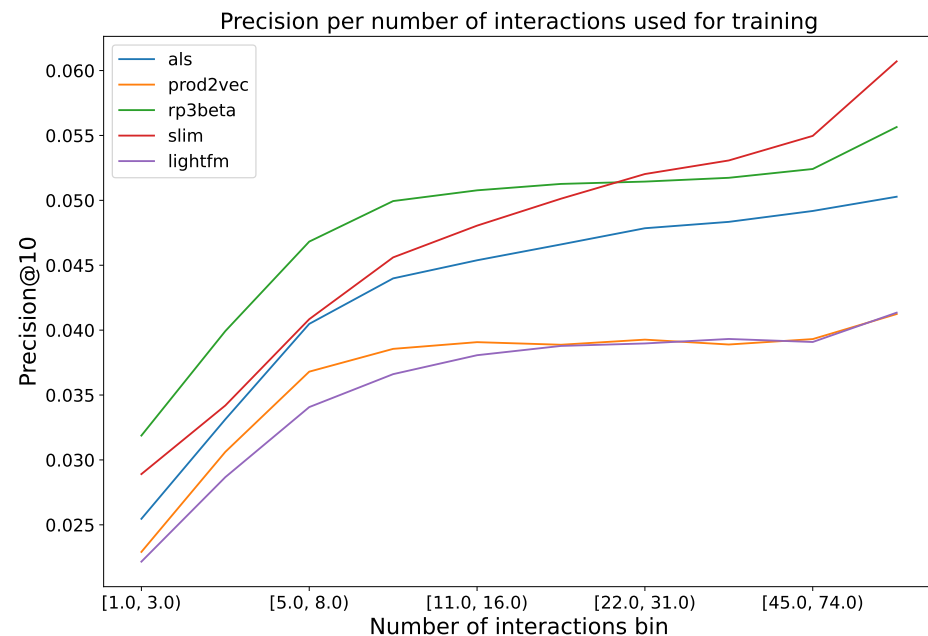


Figure 6. Precision@10 for each model, depending on the number of items with which the user interacted. It is best viewed in color. Source: own study.

We can see that the order of models sorted by precision@10 does not depend on the group, except that SLIM performed better than RP3Beta for users with at least 22 interactions. The difference is statistically significant for each group of users (discussed in Section 5.1.4). The RP3Beta model can be seen as a particular case of the item-based collaborative filtering approach, which uses a deterministic similarity measure between items [64]. On the other hand, SLIM uses machine learning methods to learn similarities. We suppose that such learned similarity scores are less biased but have higher variance, which leads to better performance for users with many interactions because averaging is done over a more significant number of scores.

5.1.4. Statistical Evaluation of Accuracy for Users with Different Numbers of Interactions

In this subsection, we discuss the difference between the RP3Beta and SLIM models depending on the number of user interactions, as observed in Figure 6.

To compare two methods over multiple datasets statistically, Demšar [62] recommends the Wilcoxon signed-rank test [65]. The Wilcoxon signed-rank test is a non-parametric alternative to the paired t test, which ranks the differences in performances of two algorithms for each dataset, ignoring the signs. It compares the ranks of the positive and negative differences.

Table 8 shows that the difference between the RP3Beta and SLIM models is statistically significant for each user group.

Table 8. Results of Wilcoxon signed-rank test between RP3Beta and SLIM models depending on the number of items with which the user interacted.

Bin	<i>p</i> -Value
[1.0, 3.0)	0
[3.0, 5.0)	0
[5.0, 8.0)	0
[8.0, 11.0)	0
[11.0, 16.0)	0
[16.0, 22.0)	1.25×10^{-6}
[22.0, 31.0)	9.27×10^{-7}
[31.0, 45.0)	0
[45.0, 74.0)	0
[74.0, 852.0)	0

Source: own study.

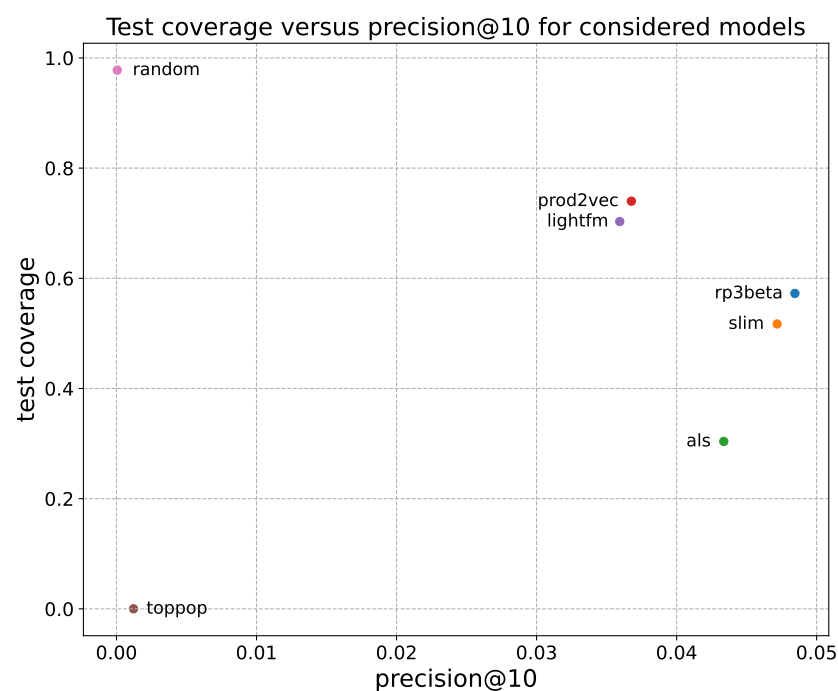
5.2. Diversity

In Table 9, we can observe that Prod2Vec and LightFM provide the greatest diversity after excluding random recommendations (which should have the greatest diversity). The diversities of the most accurate models, RP3Beta, SLIM, and ALS, are similar, except for the case of the lower test coverage of ALS. The relation between test coverage and precision@10 is shown in Figure 7.

Table 9. Diversity: evaluation results.

Metric	RP3Beta	SLIM	ALS	Prod2Vec	LightFM	Most Popular	Random
test coverage	0.5725	0.5171	0.3038	0.7400	0.7031	0.0002	0.9778
Shannon	9.5271	9.6728	9.6270	10.4031	10.1385	2.3296	11.7267
Gini	0.9083	0.9029	0.9120	0.7956	0.8397	0.9999	0.1159

Source: own study.

**Figure 7.** Test coverage versus precision@10 for considered models. Source: own study.

5.3. Overlap of Methods

In our research, we focus on methods that are worth testing with real users. Therefore, we studied the overlap between recommendations offered by the implemented methods concerning the user–item pairs (Overlap Coefficient [66]). We expect more significant differences in recommendations when comparing methods with a low overlap.

The results of this experiment are reported in Table 10. RP3Beta and SLIM offer similar recommendations. Although they are calculated differently, they use a sparse item similarity matrix. The comparison group with ALS has medium similarities with other methods. The most different from the overlap perspective are also the most diverse models, namely Prod2Vec and LightFM. The low overlap of ALS and LightFM emphasizes the importance of choosing the loss function in matrix factorization approaches.

Table 10. Overlap of models.

Model	RP3Beta	SLIM	ALS	Prod2Vec	LightFM
RP3Beta	100%	73%	53%	37%	38%
SLIM	73%	100%	50%	35%	35%
ALS	53%	50%	100%	38%	37%
Prod2Vec	37%	35%	38%	100%	28%
LightFM	38%	35%	37%	28%	100%

Source: own study.

5.4. Evaluation of Efficiency

The results of the efficiency evaluation are presented in Figure 8. RP3Beta, ALS, and LightFM outperform SLIM and Prod2Vec regarding execution time, whereas Prod2Vec requires the least memory. To retrain the model frequently, we set up the maximal acceptable execution time as 3 h, which was satisfied by ALS, LightFM, and RP3Beta. From a cost perspective, the acceptable memory utilization was set up to 110 GB, which all models fulfilled. We can observe that our models could be trained even on a single laptop.

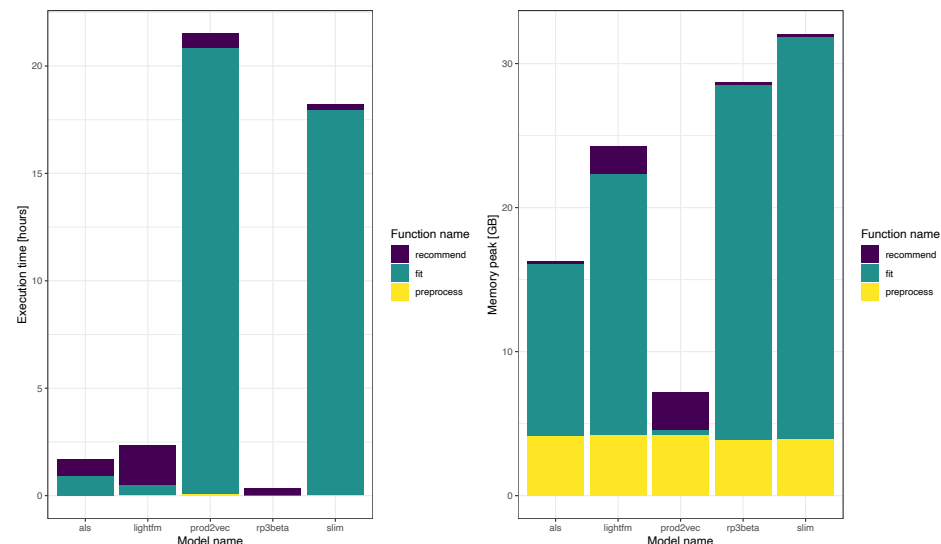


Figure 8. Execution time and memory utilization. It is best viewed in color. Source: own study.

5.5. Summary

The laboratory research enabled us to select models to be tested online with OLX users. Firstly, we chose the RP3Beta model for online tests because it is the best approach regarding accuracy and execution time. The relatively high memory consumption is still low enough for the model to be tested in production. Investigating the impact of this method's relatively low diversity may be an exciting line of future research.

SLIM is the second-best model in terms of precision@10. Nevertheless, we did not select this method for online evaluation due to its low efficiency and high overlap with the already selected RP3Beta model. Improving the efficiency of the SLIM implementation might change this decision.

Among other models, the highest precision@10 is achieved by ALS. Additionally, ALS was already implemented and used for job recommendations at OLX; therefore, we decided to include it in the online comparison.

Even though Prod2Vec and LightFM produce the most diverse recommendations, we decided not to test them online due to the significantly worse precision@10 and efficiency with Prod2Vec.

6. Online Evaluation

To evaluate the effectiveness of the selected methods, we conducted two online A/B tests with OLX Jobs users. We focused on users who had recently visited job ads on our platform. We sent them emails with ten recommended job ads, and when a user had installed an OLX application, a push notification with one recommended job ad was also sent. Then, we observed whether the user applied for any job within the next 48 h (converted user).

6.1. Impact of the ALS Model

The first experiment aimed to answer the question: does the introduction of the recommendation method increase the number of users applying for jobs? In this test, we split the users into two groups: a control group without recommendations and a group with recommendations generated using the ALS model. Only $\approx 10\%$ of users were assigned to the control group because the ALS model was already implemented and used for job recommendations at OLX before the test. We preferred to gather a minimal sample by running a more extended test to observe a longer-term impact on our users. The test was carried out for 25 days in March 2021. The results are reported in Table 11. The statistical significance of the advantage of ALS over the control group was examined using the chi-squared test, and the p -value was found to equal 0. This experiment confirmed our hypothesis that recommendations affect activity, as the percentage of converted users after receiving a communication increased significantly by $(16.83\% - 15.98\%)/15.98\% \approx 5.3\%$. The observed rise translates into hundreds of additional users applying for jobs through OLX Jobs daily.

Table 11. Comparison of activity with or without recommendations.

Variant	Users	% Converted Users
control	129,308	15.98%
ALS	1,170,262	16.83%

Source: own study.

6.2. Comparison of ALS and RP3Beta Models

In the second experiment, we tested whether the best model from laboratory testing outperformed ALS. We split our users into three groups, receiving recommendations from different recommendation systems: ALS, RP3Beta, and a mixed variant. We generated half of the recommendations with ALS and the other half with RP3Beta. The second experiment was conducted for 28 days in March and April 2021. The results are presented in Table 12. The percentage of converted users is calculated here relative to all users; however, changing the recommendation system did not affect users who did not open the message. We decided to filter these out, as their behavior added unnecessary noise to our data. Table 13 presents the impact on users who opened the message. The advantage of RP3Beta over ALS is statistically significant (p -value $\approx 10^{-6}$). The difference between the variants RP3Beta and ALS+RP3Beta is not statistically significant (the p -value is 0.19).

Table 12. Comparison of different recommendation methods.

Variant	Users	% Converted Users
ALS	343,892	15.25%
RP3Beta	345,273	15.40%
ALS+RP3Beta	343,896	15.30%

Source: own study.

Table 13. Comparison of different recommendation methods for users who opened the message.

Variant	Users	% Converted Users
ALS	44,775	19.66%
RP3Beta	46,097	20.94%
ALS+RP3Beta	45,469	20.59%

Source: own study.

Out of the users who opened the message, we observed 1.28 pp (percentage points) more converted users from RP3Beta than from ALS. Since the message open rate was 13.2%, this is equivalent to an increase of $1.28 \times 0.132 \approx 0.17$ pp of all targeted users. Based on the first experiment's results, ALS brings a 0.85 pp increase over the control group. This means that replacing ALS with RP3Beta increases our impact on all target users by approximately $0.17/0.85 = 20\%$.

6.3. Discussion

We have proved that sending job recommendations from the ALS model increases the number of users responding to job ads by more than 5%. Moreover, RP3Beta significantly outperforms ALS.

Additionally, we carried out another online A/B test comparing the RP3Beta model with a variant where we replaced half of the RP3Beta recommendations with recommendations generated from the deep learning hybrid recommender system utilized at OLX (<https://tech.olx.com/item2vec-neural-item-embeddings-to-enhance-recommendations-1fd948a6f293>, accessed on 29 May 2024). We decided not to describe that test in this paper, as this work focuses on collaborative filtering approaches. We also wanted to keep our results reproducible on the published dataset. Unfortunately, we could not publish all of the features utilized by our internal model. We did not observe any improvements over RP3Beta, even though the second model utilizes additional knowledge about job ads. Hence, we confirmed that a properly determined and tuned simple model might outperform more advanced deep solutions. This is consistent with the results from Dacrema et al. [64] and Anelli et al. [30], who compared several non-neural approaches, including RP3Beta, with more advanced neural models.

The efficiency of the RP3Beta model enables OLX Jobs to train it from scratch multiple times per day with very low costs (on a CPU). The state-of-the-art collaborative filtering model at OLX jobs is currently used to generate email and push recommendations for our users. We recently adapted this model to serve on-site recommendations in real-time, considering recent user interactions with items [67]. We also proposed the P3 learning to rank (P3LTR) model [68], a generalization of the RP3Beta, where we learn the importance of user–item relations based on types of user–item past interactions and timestamps.

In this work, we have addressed the research gap of the lack of comparisons of classical collaborative filtering approaches in the job domain by providing comprehensive laboratory research on a real-world dataset and an online evaluation with millions of users. Additionally, we have made the source code for offline assessment and the dataset publicly available. We encourage other researchers to develop new recommendation methods to help users find their ideal job more quickly.

The methods considered in this work only require (user, item, and timestamp) triplets as input data for training and evaluation; hence, they can be applied across several domains,

even for datasets consisting of tens of millions of interactions. We expect good results in the classifieds domain based on our initial results obtained in other categories of OLX.

7. Summary

This paper has addressed the topic of job recommendations in online classifieds, using the example of OLX Jobs. We performed an extensive evaluation of approaches representing different families of recommendation methods.

During the laboratory research, we demonstrated that RP3Beta, SLIM, and ALS perform significantly better than LightFM and Prod2Vec. We also found that Prod2Vec offers the most extraordinary diversity, which may be necessary when working on the novelty of recommendations. Moreover, we examined how similar the recommendations generated by different models are.

We also conducted an evaluation by sending millions of messages to online users. A/B tests were carried out using ALS and RP3Beta. The results of these tests demonstrated that sending job recommendations generated by these models increases the number of users contacting advertisers from OLX Jobs. With RP3Beta available in the production environment, we improved our effectiveness by 20% (when considering user reactions).

When evaluating the methods from the classifieds perspective, a model recommending the most popular (as yet unseen) items was unsuitable for job recommendations. A recommendation model based on random recommendations was also added for comparison to demonstrate the advantage of the researched methods.

Another significant result of this research is the dataset, which will enable further research on recommendations for classified needs.

One of the weaknesses of our work is a lack of comparison against deep neural networks (DNNs) or graph neural networks (GNNs). For such methods, satisfying our time constraints without more expensive infrastructure (e.g., with GPU) is challenging. Relaxing our constraints or comparing the most efficient DNNs or GNNs would be interesting for future work. Another weakness is that the comparison is only on a single dataset. The size of the dataset has a significant impact on method selection. It would be interesting to benchmark the same methods on datasets of different sizes, either subsets of the original dataset or other datasets from the job domain (e.g., from different markets the OLX operates). Additionally, it would be interesting to explore the correlation between diversity and online results, aiming to develop an offline metric that better aligns with business goals.

Author Contributions: Conceptualization, R.K.; methodology, R.K. and T.G.; software, R.K. and V.D.; validation, R.K., V.D., T.G. and A.F.; formal analysis, R.K. and T.G.; investigation, R.K. and T.G.; resources, R.K.; data curation, R.K. and V.D.; writing—original draft preparation, R.K., A.F. and T.G.; writing—review and editing, R.K., A.F. and T.G.; visualization, R.K. and T.G.; supervision, T.G. and A.F.; project administration, R.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Dataset: <https://www.kaggle.com/datasets/olxdatascience/olx-jobs-interactions>, accessed on 29 May 2024; source code: <https://github.com/rob-kwiec/olx-jobs-recommendations>, accessed on 29 May 2024

Conflicts of Interest: Authors Robert Kwieciński, Agata Filipowska and Viacheslav Dubrov were employed by the company OLX Group. The remaining author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Davis, M. What are Online Classifieds? *Big Easy Magazine*, 4 January 2021.
2. Jones, S. *Online Classifieds*; Pew Internet & American Life Project: Washington, DC, USA, 2009.
3. Gourville, J.; Soman, D. Overchoice and Assortment Type: When and Why Variety Backfires. *Mark. Sci.* **2005**, *24*, 382–395. [CrossRef]

4. Chernev, A. When More Is Less and Less Is More: The Role of Ideal Point Availability and Assortment in Consumer Choice. *J. Consum. Res.* **2003**, *30*, 170–183. [\[CrossRef\]](#)
5. Boatwright, P.; Nunes, J. Reducing Assortment: An Attribute-Based Approach. *J. Mark.* **2001**, *65*, 50–63. [\[CrossRef\]](#)
6. Gómez-Urbe, C.; Hunt, N. The Netflix Recommender System. *ACM Trans. Manag. Inf. Syst.* **2015**, *6*, 1–19. [\[CrossRef\]](#)
7. Smith, B.; Linden, G. Two Decades of Recommender Systems at Amazon.com. *IEEE Internet Comput.* **2017**, *21*, 12–18. [\[CrossRef\]](#)
8. Aggarwal, C.C. *Recommender Systems: The Textbook*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2016.
9. Li, S.; Karahanna, E. Online Recommendation Systems in a B2C E-Commerce Context: A Review and Future Directions. *J. Assoc. Inf. Syst.* **2015**, *16*, 72–107. [\[CrossRef\]](#)
10. Twardowski, B. Modelling Contextual Information in Session-Aware Recommender Systems with Neural Networks. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 273–276. [\[CrossRef\]](#)
11. Alotaibi, S. A survey of job recommender systems. *Int. J. Phys. Sci.* **2012**, *7*, 5127–5142. [\[CrossRef\]](#)
12. Abel, F.; Benczúr, A.; Kohlsdorf, D.; Larson, M.; Pálovics, R. *RecSys Challenge'16: Proceedings of the Recommender Systems Challenge*; Association for Computing Machinery: New York, NY, USA, 2016.
13. Abel, F.; Deldjoo, Y.; Elahi, M.; Kohlsdorf, D. *RecSys Challenge'17: Proceedings of the Recommender Systems Challenge 2017*; Association for Computing Machinery: New York, NY, USA, 2017.
14. Li, S.; Shi, B.; Yang, J.; Yan, J.; Wang, S.; Chen, F.; He, Q. Deep Job Understanding at LinkedIn. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 25–30 July 2020; pp. 2145–2148. [\[CrossRef\]](#)
15. Zhu, G.; Chen, Y.; Wang, S. Graph-Community-Enabled Personalized Course-Job Recommendations with Cross-Domain Data Integration. *Sustainability* **2022**, *14*, 7439. [\[CrossRef\]](#)
16. Lacic, E.; Reiter-Haas, M.; Kowald, D.; Reddy Dareddy, M.; Cho, J.; Lex, E. Using autoencoders for session-based job recommendations. *User Model. User-Adapt. Interact.* **2020**, *30*, 617–658. [\[CrossRef\]](#)
17. Pazzani, M.J.; Billsus, D. Content-Based Recommendation Systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 325–341. [\[CrossRef\]](#)
18. Chen, R.; Hua, Q.; Chang, Y.S.; Wang, B.; Zhang, L.; Kong, X. A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks. *IEEE Access* **2018**, *6*, 64301–64320. [\[CrossRef\]](#)
19. Wang, X.; He, X.; Wang, M.; Feng, F.; Chua, T.S. Neural Graph Collaborative Filtering. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 165–174. [\[CrossRef\]](#)
20. Schafer, J.B.; Frankowski, D.; Herlocker, J.; Sen, S. Collaborative filtering recommender systems. In *The Adaptive Web*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 291–324. [\[CrossRef\]](#)
21. Takács, G.; Pilászy, I.; Németh, B.; Tikk, D. Scalable Collaborative Filtering Approaches for Large Recommender Systems. *J. Mach. Learn. Res.* **2009**, *10*, 623–656. [\[CrossRef\]](#)
22. Jawaheer, G.; Szomszor, M.; Kostkova, P. Comparison of implicit and explicit feedback from an online music recommendation service. In Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, Barcelona, Spain, 26 September 2010; pp. 47–51. [\[CrossRef\]](#)
23. Tamm, Y.M.; Damdinov, R.; Vasilev, A. Quality Metrics in Recommender Systems: Do We Calculate Metrics Consistently? In Proceedings of the Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September–1 October 2021; pp. 708–713. [\[CrossRef\]](#)
24. Kaminskas, M.; Bridge, D. Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems. *ACM Trans. Interact. Intell. Syst.* **2016**, *7*, 1–42. [\[CrossRef\]](#)
25. Kunaver, M.; Pozrl, T. Diversity in Recommender Systems, A Survey. *Knowl.-Based Syst.* **2017**, *123*, 154–162. [\[CrossRef\]](#)
26. Ge, M.; Delgado-Battenfeld, C.; Jannach, D. Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity. In Proceedings of the Fourth ACM Conference on Recommender Systems, Barcelona, Spain, 26–30 September 2010; pp. 257–260. [\[CrossRef\]](#)
27. Mogenet, A.; Pham, T.A.N.; Kazama, M.; Kong, J. Predicting Online Performance of Job Recommender Systems with Offline Evaluation. In Proceedings of the 13th ACM Conference on Recommender Systems, Copenhagen, Denmark, 16–20 September 2019; pp. 477–480. [\[CrossRef\]](#)
28. Anelli, V.W.; Bellogín, A.; Di Noia, T.; Jannach, D.; Pomo, C. Top-N Recommendation Algorithms: A Quest for the State-of-the-Art. In Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization, Barcelona, Spain, 4–7 July 2022; pp. 121–131. [\[CrossRef\]](#)
29. Dacrema, M.F.; Boglio, S.; Cremonesi, P.; Jannach, D. A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research. *ACM Trans. Inf. Syst.* **2021**, *39*, 1–49. [\[CrossRef\]](#)
30. Anelli, V.W.; Bellogín, A.; Di Noia, T.; Pomo, C. Reenvisioning the Comparison between Neural Collaborative Filtering and Matrix Factorization. In Proceedings of the Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September–1 October 2021; pp. 521–529. [\[CrossRef\]](#)

31. Zhu, J.; Dai, Q.; Su, L.; Ma, R.; Liu, J.; Cai, G.; Xiao, X.; Zhang, R. BARS: Towards Open Benchmarking for Recommender Systems. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 2912–2923. [\[CrossRef\]](#)
32. Dong, Y.; Li, J.; Schnabel, T. When Newer is Not Better: Does Deep Learning Really Benefit Recommendation From Implicit Feedback? In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, Taipei, Taiwan, 23–27 July 2023; pp. 942–952. [\[CrossRef\]](#)
33. Anelli, V.W.; Malitesta, D.; Pomo, C.; Bellogin, A.; Di Sciascio, E.; Di Noia, T. Challenging the Myth of Graph Collaborative Filtering: A Reasoned and Reproducibility-driven Analysis. In Proceedings of the 17th ACM Conference on Recommender Systems, Singapore, 18–22 September 2023; pp. 350–361. [\[CrossRef\]](#)
34. Koren, Y.; Bell, R.; Volinsky, C. Matrix Factorization Techniques for Recommender Systems. *Computer* **2009**, *42*, 30–37. [\[CrossRef\]](#)
35. Kula, M. Metadata embeddings for user and item cold-start recommendations. *arXiv* **2015**, arXiv:1507.08439.
36. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian Personalized Ranking from Implicit Feedback. *arXiv* **2012**, arXiv:1205.2618
37. Weston, J.; Bengio, S.; Usunier, N. Wsabie: Scaling up to large vocabulary image annotation. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011; pp. 2764–2770.
38. Weston, J.; Yee, H.; Weiss, R.J. Learning to Rank Recommendations with the K-Order Statistic Loss. In Proceedings of the 7th ACM Conference on Recommender Systems, Hong Kong, China, 12–16 October 2013; pp. 245–248. [\[CrossRef\]](#)
39. Hu, Y.; Koren, Y.; Volinsky, C. Collaborative Filtering for Implicit Feedback Datasets. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 263–272. [\[CrossRef\]](#)
40. Takács, G.; Pilászy, I.; Tikk, D. Applications of the Conjugate Gradient Method for Implicit Feedback Collaborative Filtering. In Proceedings of the Fifth ACM Conference on Recommender Systems, Chicago, IL, USA, 23–27 October 2011; pp. 297–300. [\[CrossRef\]](#)
41. Ning, X.; Karypis, G. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In Proceedings of the 2011 IEEE 11th International Conference on Data Mining, Vancouver, BC, Canada, 11–14 December 2011; pp. 497–506. [\[CrossRef\]](#)
42. Christoffel, F.; Paudel, B.; Newell, C.; Bernstein, A. Blockbusters and Wallflowers: Accurate, Diverse, and Scalable Recommendations with Random Walks. In Proceedings of the 9th ACM Conference on Recommender Systems, Vienna, Austria, 16–20 September 2015; pp. 163–170. [\[CrossRef\]](#)
43. Cooper, C.; Lee, S.H.; Radzik, T.; Siantos, Y. Random Walks in Recommender Systems: Exact Computation and Simulations. In Proceedings of the 23rd International Conference on World Wide Web, New York, NY, USA, 7–11 April 2014; pp. 811–816. [\[CrossRef\]](#)
44. Grbovic, M.; Radosavljevic, V.; Djuric, N.; Bhamidipati, N.; Savla, J.; Bhagwan, V.; Sharp, D. E-Commerce in Your Inbox: Product Recommendations at Scale. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, 10–13 August 2015; pp. 1809–1818. [\[CrossRef\]](#)
45. Barkan, O.; Koenigstein, N. ITEM2VEC: Neural item embedding for collaborative filtering. In Proceedings of the 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), Vietri sul Mare, Italy, 13–16 September 2016; pp. 1–6. [\[CrossRef\]](#)
46. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems—Volume 2, Red Hook, NY, USA, 5–10 December 2013; pp. 3111–3119.
47. De Ruijt, C.; Bhulai, S. Job recommender systems: A review. *arXiv* **2021**, arXiv:2111.13576.
48. Kula, M. LightFM—Github Repository. 2023. Available online: <https://github.com/lyst/lightfm> (accessed on 29 May 2024).
49. Frederickson, B. ALS—Github Repository. 2022. Available online: <https://github.com/benfred/implicit/blob/main/implicit/als.py> (accessed on 29 May 2024).
50. Levy, M.; Grisel, O. SLIM—Github Repository. 2014. Available online: https://github.com/Mendeley/mrec/blob/master/mrec/item_similarity/slim.py (accessed on 29 May 2024).
51. Quadrana, M. SLIM—Github Repository. 2021. Available online: https://github.com/MaurizioFD/RecSys2019_DeepLearning_Evaluation/blob/master/SLIM_ElasticNet/SLIMElasticNetRecommender.py (accessed on 29 May 2024).
52. Bernardis, C. RP3Beta—Github Repository. 2021. Available online: https://github.com/MaurizioFD/RecSys2019_DeepLearning_Evaluation/blob/master/GraphBased/RP3betaRecommender.py (accessed on 29 May 2024).
53. Paudel, B.; Christoffel, F.; Newell, C.; Bernstein, A. Updatable, Accurate, Diverse, and Scalable Recommendations for Interactive Applications. *ACM Trans. Interact. Intell. Syst.* **2016**, *7*, 1–34. [\[CrossRef\]](#)
54. Řehůřek, R.; Sojka, P. Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, 22 May 2010; pp. 45–50, ISBN 2-9517408-6-7. Available online: <http://is.muni.cz/publication/884893/en> (accessed on 29 May 2012).
55. Meng, Z.; McCreddie, R.; Macdonald, C.; Ounis, I. Exploring Data Splitting Strategies for the Evaluation of Recommendation Models. In Proceedings of the 14th ACM Conference on Recommender Systems, Virtual, 22–26 September 2020; pp. 681–686. [\[CrossRef\]](#)
56. Shani, G.; Gunawardana, A., Evaluating Recommendation Systems. In *Recommender Systems Handbook*; Springer: Boston, MA, USA, 2011; pp. 257–297. [\[CrossRef\]](#)

57. Friedman, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [[CrossRef](#)]
58. Friedman, M. A Comparison of Alternative Tests of Significance for the Problem of m Rankings. *Ann. Math. Stat.* **1940**, *11*, 86–92. [[CrossRef](#)]
59. Iman, R.L.; Davenport, J.M. Approximations of the critical region of the Friedman statistic. *Commun. Stat.-Theory Methods* **1980**, *9*, 571–595. [[CrossRef](#)]
60. García, S.; Herrera, F. An extension on “Statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons. *J. Mach. Learn. Res.* **2008**, *9*, 2677–2694.
61. García, S.; Fernández, A.; Luengo, J.; Herrera, F. Advanced Nonparametric Tests for Multiple Comparisons in the Design of Experiments in Computational Intelligence and Data Mining: Experimental Analysis of Power. *Inf. Sci.* **2010**, *180*, 2044–2064. [[CrossRef](#)]
62. Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
63. Nemenyi, P. Distribution-Free Multiple Comparisons. Ph.D. Thesis, Princeton University, Princeton, NJ, USA, 1963.
64. Dacrema, M.F.; Cremonesi, P.; Jannach, D. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. In Proceedings of the 13th ACM Conference on Recommender Systems, Copenhagen, Denmark, 16–20 September 2019; pp. 101–109. [[CrossRef](#)]
65. Wilcoxon, F. Individual Comparisons by Ranking Methods. *Biom. Bull.* **1945**, *1*, 80–83. [[CrossRef](#)]
66. Vijaymeena, M.; Kavitha, K. A Survey on Similarity Measures in Text Mining. *Mach. Learn. Appl. Int. J.* **2016**, *3*, 19–28. [[CrossRef](#)]
67. Kwieciński, R.; Melniczak, G.; Górecki, T. Comparison of Real-Time and Batch Job Recommendations. *IEEE Access* **2023**, *11*, 20553–20559. [[CrossRef](#)]
68. Kwieciński, R.; Górecki, T.; Filipowska, A. Learning edge importance in bipartite graph-based recommendations. In Proceedings of the 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS), Sofia, Bulgaria, 4–7 September 2022; pp. 227–233. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Blockchain and Homomorphic Encryption for Data Security and Statistical Privacy

Rahul Raj ^{1,*}, Yeşem Kurt Peker ¹ and Zeynep Delal Mutlu ²¹ TSYS School of Computer Science, Columbus State University, Columbus, GA 31907, USA; peker_yesem@columbusstate.edu² Independent Researcher, 1200 Brussels, Belgium; zeynepdelalmutlu@gmail.com

* Correspondence: raj_rahul@students.columbusstate.edu

Abstract: This study proposes a blockchain-based system that utilizes fully homomorphic encryption to provide data security and statistical privacy when data are shared with third parties for analysis or research purposes. The proposed system not only provides security of data in transit, at rest, and in use but also assures privacy and computational integrity for simple statistical computations. This is achieved by leveraging the attributes of the blockchain technology, which provides availability and data integrity, combined with homomorphic encryption, which provides confidentiality of data in use. The computations are performed on smart contracts residing on the blockchain, providing computational integrity. The proposed system is implemented on the Zama blockchain and performs statistical operations including mean, median, and variance on encrypted data. The results indicate that it is possible to perform fully homomorphic computations on the blockchain. Even though current computing limitations on the blockchain do not allow running the system for large data sets, the technology is available, and with advancements toward more efficient homomorphic operations on blockchains, the proposed system will provide an ultimate solution for providing the much-desired security properties in applications, including data and statistical privacy, confidentiality, and integrity at rest, in transit, and in use.

Keywords: data privacy; data integrity; computational integrity; homomorphic encryption; blockchain; statistical privacy; data security; statistical confidentiality



Citation: Raj, R.; Kurt Peker, Y.; Mutlu, Z.D. Blockchain and Homomorphic Encryption for Data Security and Statistical Privacy. *Electronics* **2024**, *13*, 3050. <https://doi.org/10.3390/electronics13153050>

Academic Editor: Simeone Marino

Received: 8 June 2024

Revised: 20 July 2024

Accepted: 26 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Protecting the privacy of users and the confidentiality of sensitive data is vital in all systems where user data are stored, transmitted, or processed. Often, data need to be analyzed for business purposes or for the benefit of the public. For example, census data record information about the population of a country, state, city, or other well-defined geographical areas. They include sensitive information such as age, race, gender, religious beliefs, and long-term health conditions in many cases. Census data are the primary data used to understand the social, economic, and demographic conditions locally and nationally and provide information that governments need to develop policies, plan and run public services, and allocate funding. The protection of census data is essential from collection to processing. One safeguard is to allow only authorized people access to the data. Encryption is a common mechanism to implement this safeguard. Data are encrypted in storage and transmitted between the authorized users. However, there is the risk of unintentional mistakes by authorized users or an authorized user turning malicious, jeopardizing the security and privacy of the data and the integrity of the analysis. In some cases, data need to be made available to third parties (e.g., researchers) so that proper and objective analysis can be performed. In these cases, to protect user privacy, data are de-identified or anonymized before they are shared. However, these methods do not always preserve user privacy as intended. In addition to unintentional mistakes and the risk of insider

threats, when large data sets are made available, user information may be leaked due to more sophisticated attacks. especially when large data sets are made available. The 2016 study “Exposed! A Survey of Attacks on Private Data” by Dwork et al. presents a survey of common attacks against aggregate data. It describes reconstruction and tracing attacks. Reconstruction attacks involve approximating sensitive features of individuals within a data set. Tracing attacks aim to determine whether a target individual’s data are included in the data set or not [1]. Dwork’s study is followed by many studies focusing on attacks to data privacy and mechanisms to defend against them [2–6].

Two terms related to security of data in the statistical sense are statistical privacy and statistical confidentiality. Statistical privacy is “the art of designing a privacy mechanism that transforms sensitive data into data that are simultaneously useful and non-sensitive” [7]. The goal of statistical privacy methodologies is to “enable broad sharing of data across different data contexts and domains where it is desired or required that individuals’ identities or sensitive attributes are protected (e.g., census, health, genomic data, social networks)” [8].

Statistical confidentiality refers to the “protection of information of individual statistical units”, or equivalently, “the protection of data collected for statistical purposes” [9]. Statistical confidentiality is closely related to statistical privacy. Even though no clear distinction between these terms exists in the literature, based on the definitions referenced in this article, we consider statistical confidentiality to mean the protection of the statistical units, i.e., data collected on individuals or businesses, and ensuring the approved use of them. Statistical privacy is the broader security concept that includes the transformation of sensitive data into non-sensitive and useful data. Statistical confidentiality is one of the fundamental principles of official statistics endorsed by the General Assembly of the United Nations [10]. It is also a fundamental principle of European statistics, defining principles, rules, and procedures to protect confidential data while still permitting their use for statistical purposes [11].

The first publication of Innovations in Federal Statistics [12] discusses the challenges of and the need for a new paradigm in the federal statistical system. Chapter 5 of the publication titled “Protecting Privacy and Confidentiality While Providing Access to Data for Research Use” focuses on protecting the confidentiality of the information collected for statistical use. After discussing the security implications of linking multiple data sets for analysis, including the legal foundation for privacy and confidentiality, the authors draw two major conclusions: “As federal statistical agencies move forward with linking multiple data sets, they must simultaneously address quantifying and controlling the risk of privacy loss” and “Privacy-enhancing techniques and privacy-preserving statistical data analysis can potentially enable the use of private-sector data sources for federal statistics”. With these conclusions, they recommend “Federal statistical agencies should adopt modern database, cryptography, privacy-preserving, and privacy-enhancing technologies”.

In this study, we propose a system that provides data security, statistical privacy, and confidentiality using techniques and tools purely from mathematics and computer science. We explore the use of a blockchain-based system that allows performing statistical calculations on encrypted data by utilizing fully homomorphic encryption on the blockchain using smart contracts. The proposed system provides end-to-end security of the data from the time they leave the data owner to the point when the researcher receives the results of statistical computations. The provided security includes the integrity and confidentiality of the data in transit, at rest, and in use. Not only the individual data points are never revealed in plaintext, but the results are available only to authorized users in plaintext form. Also, because the computations are carried out on the blockchain, tampering with the algorithms for the statistical computations is almost impossible. Moreover, requesting statistics other than what is approved on the smart contract is not possible. As such, the proposed system provides statistical privacy and confidentiality desired in various applications, including federal statistical systems. In Section 2, we provide brief descriptions of recent studies where blockchain and homomorphic encryption are used to provide data security. In

Section 3, we describe our proposed system and the implementation of a proof of concept for the system. In Section 4, we present the results of running the proof of concept, and finally, in Section 5, we provide a discussion of the results.

2. Related Work

There have been numerous studies that have attempted to introduce data security and integrity not only when they are at rest but also when they are in use through various means, including homomorphic encryption and blockchain technology. One such study proposes a system for a collaborative data training paradigm for medical image data sharing where the blockchain ledger provides the decentralization of the federated learning models without relying on a central server, and the homomorphic encryption alleviates the concerns related to raw data sharing [13]. Their system encrypts the gradients homomorphically and shares them through the blockchain.

Liang et al. [14] propose the integration of blockchain and homomorphic encryption to address the challenges in circuit copyright protection. Their study proposes a homomorphic encryption-based mathematical model within the blockchain that secures the transactions while also ensuring integrity and confidentiality of data by utilizing smart contracts.

Yaji et al. [15] proposed a system that utilizes Goldwasser–Micali and Paillier encryption schemes for the comparative evaluation study with a focus on data privacy techniques using blockchain technology for AI applications. The study found that attacks on the blockchain such as collision, preimage, and attacks on the wallet can be avoided through encrypting blocks using the proposed Goldwasser–Micali and Paillier encryption schemes.

Mutlu et al. [16] proposed a system that uses blockchain technology and homomorphic encryption and enables third parties (researchers) to perform linear regression on encrypted data. They use the Paillier algorithm to calculate the sum required for linear regression through smart contracts. The data owner encrypts the data using the public key of the researcher and sends them to the smart contract where the calculation is performed. The encrypted result can then be accessed by the researcher, who can decrypt it on their system using their private key.

Vanin et al. [17] propose a model to secure Personal Health Record (PHR) that uses an interplanetary protocol file system based on distributed hash tables (DHTs) along with the blockchain. PHR metadata are stored on the blockchain and shared across the network, while PHR data are stored off-chain through the IPFS network. They use two elements: Data Steward (DS), which is responsible for storing PHR on behalf of the individual, and Shared Data Vault (SDV), which is a temporary IPFS storage area where health institutions can access PHR with the consent of the individual. Encrypted data are available to the public through statistical portals, where they can perform operations on the data using homomorphic encryption to obtain meaningful results. For this purpose, they use the Microsoft Simple Encrypted Arithmetic Library (SEAL), which implements the BFV algorithm in JavaScript.

Umar et al. [18] proposed a model for e-voting using the Paillier algorithm. Once the voter casts their vote, it is encrypted homomorphically. A new block of the transaction is then created, which contains the encrypted ballots, the pseudonymous address of the voter and the admin, the timestamp of the block creation, the hash of the previous block of the transaction, as well as the hash of the current block. Then the new block of the transaction is mined using the consensus mechanism. After mining, the new block is committed to the ledger of the blockchain. This process continues until the end of the election, after which the admin tallies the encrypted votes using the Paillier algorithm, which yields the final sum, which can then be decrypted to determine the results.

Shrestha et al. [19] conducted a study that analyzes the security concerns with the Internet of Things (IoT). One of the major concerns that has been highlighted is the privacy of data. This study also explores the possibility of integrating the IoT with the blockchain and with homomorphic encryption. The benefits include data immutability, unforgettability, removing single points of failure, and confidentiality of data.

Caldarola et al. [20] proposed the Neural Fairness Protocol, a consensus mechanism that integrates the Elliptic Curve Digital Signature Algorithm (ECDSA) and cryptographic techniques. This protocol enhances anonymity and accountability in blockchain transactions, particularly through the development of a robust threshold ECDSA algorithm. Beyond its initial application, this innovation shows potential for improving security across various cryptocurrencies, such as Litecoin and Mastercoin.

The proposed system in this study can be compared to the work proposed by Liang [14], Yaji [15], and Mutlu [16] in the sense that it utilizes blockchain technology along with homomorphic encryption to provide privacy of data while they are being used. Furthermore, the system proposed by Mutlu [14] utilizes partial homomorphic encryption over the smart contract, which is similar to the architecture of our proposed system. However, the above-mentioned studies utilize the Paillier algorithm, which is partially homomorphic, whereas this study uses a fully homomorphic system that not only allows addition but also multiplication on the encrypted data, allowing a broader range of computations to be performed securely.

3. Materials and Methods

In this section, we describe a system that allows secure data sharing and analysis between data owners and third parties. The data owners are entities or individuals that have access to the plain data. They could also be seen as sensors with the capability to process data and transmit them. The owners are authorized or give consent to have the data analyzed with the condition that privacy of the data is preserved. The third parties in the system are researchers. They could be individuals or an entity authorized to do certain types of analysis on the data. They may not have the authorization to view the data in plaintext.

3.1. Description of the Proposed System

The system encrypts the data using a homomorphic encryption scheme before they leave the data owner. The encrypted data are stored on a smart contract on the blockchain. The analyses that can be performed on the data are also stored on the smart contract. When a researcher needs to do analysis on the data, assuming that the type of analysis is available on the smart contract, the analysis is performed on the encrypted data using homomorphic computations. The results of the homomorphic computations are also encrypted. To receive the result on their end, the researcher requests the result to be re-encrypted with the researcher's key so that it can be transmitted securely to the researcher and decrypted only by the researcher. It should be noted here that the plaintext data are never revealed during the entire process, even during re-encryption. When the researcher sends their public key to the smart contract for re-encryption, the smart contract initiates a distributed decryption protocol through which parts of the result are decrypted by different validators, re-encrypted using the researcher's public key, and then combined to obtain the final re-encrypted result [21]. Figure 1 shows an overview of the system with a data owner and researcher as main actors.

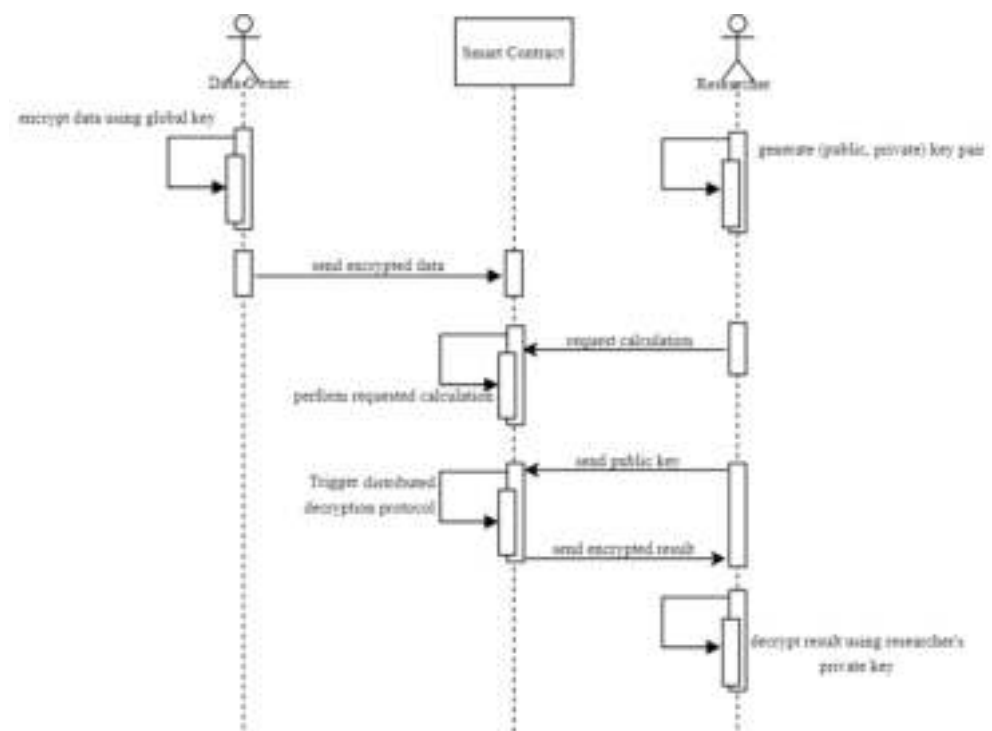


Figure 1. An overview of the proposed system.

3.2. Implementation

The proposed system is implemented on the Zama Blockchain, an Ethereum Virtual Machine (EVM)-based blockchain that supports computation on encrypted values [18]. Like Ethereum, which has a currency known as Ether, Zama has its own currency, called ZAMA. The basic idea behind Zama is to provide confidential smart contracts. This means the data sent to or received from the smart contract are encrypted and cannot be read if the data transfer is intercepted. Zama uses asymmetric encryption and hence uses two keys: public and private. The public key, also known as the global key, is stored publicly on-chain and is used by every user to encrypt their data and perform calculations on those encrypted data. The global key is generated during a setup phase by the initial validators and securely re-shared when the validator set is changed [21]. This allows mixing of encrypted data from multiple users and across multiple smart contracts. The private key is used to decrypt the data and is not owned by any single user. Instead, Zama uses a threshold protocol. In a threshold protocol, pieces of the private key are distributed among validator nodes in the network, and a certain number of validators need to cooperate to decrypt the data. In the case of Zama, the participation of at least one-third of the validators is necessary to perform decryption [18]. This method enhances security through decentralization, preventing rogue use of the key. Zama uses a probabilistic encryption scheme, which means that the encrypted value for a single plaintext will not always be the same. For example, consider an array having 4 elements [2, 4, 1, 4]. Once the elements of this array are encrypted, hypothetically, it would become [d5048, e2314, x3215, o9849]. Notice that the number 4 appears twice in the plaintext array at index 1 and index 3, but its encrypted version does not have any repetitions, and the values at index 1 and index 3 are different. This scheme further enhances security by preventing attackers from deducing information about the plaintext based on patterns in the ciphertext.

The proposed system uses the libraries for homomorphic calculations provided by Zama. Once the contract is compiled, it produces the Application Binary Interface (ABI). The ABI specifies the functions available in the smart contract along with their parameters and return types, allowing users to interact with the smart contract. The data owner and researcher both have their own JavaScript applications that utilize the ABI and methods

provided by the Ethers library to communicate with the smart contract. The data owner sends the data to the smart contract, and the researcher requests analysis from the smart contract. Figure 2 depicts the interaction of the actors with the smart contract.

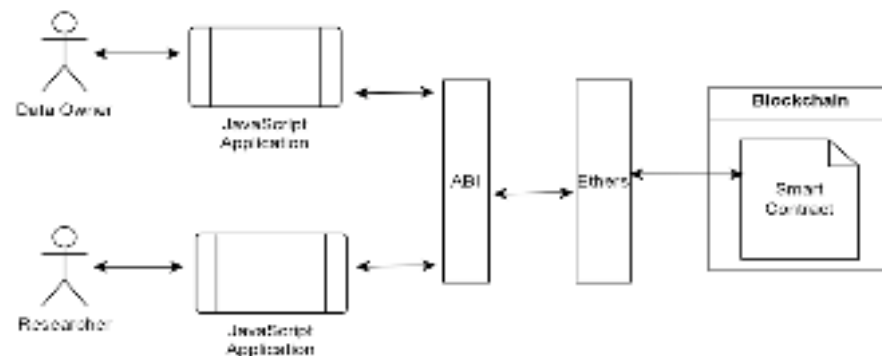


Figure 2. Interaction with the smart contract.

Following is a list of tools and technologies that were used to implement the system:

- Solidity: A programming language that is used to write smart contracts on various platforms, including Ethereum.
- Node.js: It is a JavaScript runtime environment that allows execution of JavaScript code outside the web browser. Version 18.18.0 was used for this system.
- Remix: It is a browser-based Integrated Development Environment (IDE) used for the development and deployment of smart contracts.
- MetaMask: It is a cryptocurrency wallet that is used to connect to and interact with the blockchain.
- Visual Studio Code: It is an IDE that supports development in various programming languages, including JavaScript.
- Ethers.js: It is a JavaScript library that allows interaction with smart contracts. Version 6.10.0 was used for this system.
- FhEVM: It is a library provided by Zama that allows the creation of confidential smart contracts on EVM using Solidity. Version 0.4.0 was used for this system.
- FhEVMjs: It is a JavaScript library provided by Zama that allows interaction with smart contracts, including encrypting the data and generating public/private keys. Version 0.4.0 was used for this system.

A proof of concept for the proposed system is implemented as part of this study. Due to limitations in the libraries for homomorphic computations on Zama and working on a public blockchain, as will be elaborated in Section 3.2.1, the implementation included calculations of simple descriptive statistics such as mean, median, and variance. The proof of concept was tested with a small number of data points, with the range of numbers between 1 and 50.

3.2.1. Challenges

While implementing the system, the following limitations and challenges were identified in Solidity and Zama:

- Data Types: fhEVM only supports unsigned integers that are either 8-bit, 16-bit, 32-bit, or 64-bit. Furthermore, the methods provided by fhEVM for calculations on encrypted data such as addition, subtraction, multiplication, and division return an encrypted unsigned integer. In the case of division, there is an added limitation where an encrypted integer can only be divided with a plaintext integer. This means it is not currently possible to divide two encrypted integers. Even if there was a possibility for such division, the result would always be an encrypted integer because of its return type; hence, it is not possible to work with decimal numbers. Moreover, the integers

are unsigned; therefore, it is not possible to handle negative integers either. These limitations restrict the number of analyses that can be performed on the smart contract.

- **Gas Limit:** Computations on the blockchain run on gas, a unit of cost associated with performing a transaction or computation on the network. Gas is used to pay validators for the resources they use to conduct transactions. Gas limit refers to the maximum amount of gas that can be spent on a transaction or computation. Setting a limit on gas consumption is important because it ensures that transactions have a predefined upper limit on resource consumption to safeguard against scenarios where code execution might enter an infinite loop. Currently, there is a gas limit of 10,000,000 on Zama devnet [22]. However, homomorphic calculations are quite expensive in terms of resources, and since these calculations are performed on the smart contract, they consume a lot of gas. Gas consumption depends on the operations that are performed on the encrypted data and increases with the complexity of the algorithm and bits of data type (32-bit operations would consume more gas compared to 8-bit operations). The current gas limit restricts the functionality in terms of the number of data points that can be processed.
- **Solidity:** Unlike traditional programming languages such as Python and Java, Solidity has various limitations. One of them is that the data structure used to store key-pair values (mappings) is quite limited in terms of functionality, and it is not possible to iterate over keys or values. This makes it challenging to implement various algorithms, such as the calculation of the mode (the most frequent value in an array).

3.2.2. Calculation of Mean

For a set of values $\{x_1, x_2, \dots, x_n\}$, the arithmetic mean, denoted by $\bar{x}$, is the sum of the values divided by the number of values, as shown in Equation (1).

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (1)$$

In the designed system, only the numerator, the sum of the encrypted values, is calculated on the smart contract. The division operation is performed at the researcher's end. This is due to the limitation mentioned in Section 3.2.1 that a division will always return an integer and hence lose precision in the result. To summarize, the smart contract sends the encrypted sum as well as the encrypted number of elements to the researcher. These two values are then decrypted by the researcher, and a final division is performed to obtain the result.

3.2.3. Calculation of Median

Calculating the median of sorted data is a simple task. Assuming the data consist of n values, the median is the middle element if n is odd and is the average of the two middle elements if n is even. The formula for the median of a sorted data set with n data points is given in Equation (2).

$$Med = \begin{cases} \left(\frac{n+1}{2}\right)^{th} \text{ element} & \text{if } n \text{ is odd} \\ \frac{\left(\frac{n}{2}\right)^{th} \text{ element} + \left(\frac{n}{2}+1\right)^{th} \text{ element}}{2} & \text{if } n \text{ is even} \end{cases} \quad (2)$$

In the system, the differentiation in the calculation of the median is simply handled by using an if statement. If n is odd, the median is calculated on the smart contract and returned to the researcher in encrypted form. If n is even, the sum of the middle two elements is calculated on the smart contract and sent to the researcher. The result is then decrypted by the researcher and divided by 2 to obtain the median. Thus, the researcher makes the decision of performing the division based on the number of elements.

The median calculation above assumes that the data are sorted. The biggest challenge in calculating the median of unsorted data, however, is the implementation of a sorting algorithm on the smart contract since it is not a trivial task to sort encrypted data. The

proposed system uses the bubble sort algorithm to sort the data. This algorithm was chosen because of its simplicity. It works by comparing two adjacent values in an array and swapping the numbers based on the result of that comparison, which makes it easy to implement. Furthermore, implementing bubble sort does not require any additional data structure. This makes it suitable to write in Solidity, considering the limitations of the programming language as well as the fhEVM library.

3.2.4. Calculation of Variance

The formula for calculating variance for a set of values $\{x_1, x_2, \dots, x_n\}$ is given in Equation (3), where $\bar{x}$ represents the mean of the data.

$$V = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1} \quad (3)$$

As was discussed in Section 3.2.2, it is not feasible to calculate the exact mean of any sequence on the smart contract because the division function in the fhEVM library always returns an encrypted integer. To eliminate the calculation of mean in the variance formula, we derived an equivalent formula that did not include mean or any division operator other than a major division operation at the last step. The step-by-step derivation of the formula starting with Equation (3) is shown below.

$$\begin{aligned} V &= \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1} \\ &= \frac{\sum_{i=1}^n (x_i - \frac{\sum_{j=1}^n x_j}{n})^2}{n - 1} \\ &= \frac{\sum_{i=1}^n (x_i^2 - 2x_i \frac{\sum_{j=1}^n x_j}{n} + (\frac{\sum_{j=1}^n x_j}{n})^2)}{n - 1} \\ &= \frac{n^2}{n^2} \frac{\sum_{i=1}^n (x_i^2 - 2x_i \frac{\sum_{j=1}^n x_j}{n} + \frac{(\sum_{j=1}^n x_j)^2}{n^2})}{n - 1} \\ &= \frac{\sum_{i=1}^n (n^2 x_i^2 - 2n^2 x_i \frac{\sum_{j=1}^n x_j}{n} + n^2 \frac{(\sum_{j=1}^n x_j)^2}{n^2})}{n^2(n - 1)} \\ &= \frac{n^2 \sum_{i=1}^n x_i^2 - 2n \sum_{i=1}^n (x_i \sum_{j=1}^n x_j) + \sum_{i=1}^n (\sum_{j=1}^n x_j)^2}{n^2(n - 1)} \\ &= \frac{n^2 \sum_{i=1}^n x_i^2 - 2n \sum_{i=1}^n (x_i \sum_{j=1}^n x_j) + n (\sum_{j=1}^n x_j^2)}{n^2(n - 1)} \end{aligned} \quad (4)$$

This approach removed the calculation of mean from the algorithm and, in turn, removed division as well. However, as shown in Equation (4), it was not possible to eliminate division completely. Keeping these limitations in mind, only the calculation of the numerator is implemented on the smart contract. The encrypted result of the numerator is sent to the researcher along with the encrypted value of n , where the final division is performed. Upon closer inspection of Equation (4), it can be observed that there are three major components in the numerator itself. Furthermore, the numerator involves subtraction as well. As discussed in Section 3.2.1, Zama only provides unsigned integers, which means any result below zero would undermine the accuracy of the result. To mitigate the issue, these three components are calculated separately, and their results are stored in separate variables. Then, the first component is added to the third, and the second component is then subtracted from the result of the addition. This ensures that the result does not fall below zero.

3.3. Security Assurances of the Proposed System

The proposed system provides the security requirements for all tenets of information security. It provides security assurances when data are in transit, at rest, and in use. Next, we describe how confidentiality, integrity, and availability are achieved in the proposed system.

- **Confidentiality:** Data are encrypted before they are shared and remain encrypted from then on. Data are encrypted in transit, at rest, and in use, providing statistical privacy as well as statistical confidentiality. The encrypted data are transmitted to the smart contract, where they are stored, allowing computations to be performed on them. The result of these computations is encrypted as well. Before the result is shared with the researcher, it is re-encrypted after it is internally decrypted on the blockchain using the distributed decryption protocol. This operation is not visible on the blocks of the chain and, hence, is not visible to the users. Cheating the system to obtain the decryption key would require one-third of the nodes on the blockchain to collaborate, which is an unlikely scenario for blockchains.
- **Integrity of data:** Each block in a blockchain contains the hash of the previous block, creating a chain. Tampering with one block changes the hash of that block and all the blocks that come after it. Since establishing the chain with the updated blocks is an almost impossible task, the data on the blockchain are tamper-resistant. This provides the integrity of the data.
- **Integrity of computations:** The algorithms for computations allowed on the data are implemented on the smart contract. Since smart contracts are part of the blockchain, altering them in an unauthorized way is not feasible. This provides integrity in computations and hence trust in the algorithms implemented for the computations.
- **Authorization:** Authorization in the proposed system can be considered from two perspectives. One is the control of what kind of analysis can be performed on the data. Because the functions for analysis are available on the smart contract in the proposed system and only authorized users/entities can change the functionality of the smart contract, a researcher is limited to using the functions available on the smart contract. Assuming only approved functions are implemented on the smart contract, a researcher cannot perform unapproved analysis on the data. The other authorization consideration in the system is about who can send data to the smart contract and who can request analysis of the data. Blockchain applications can be programmed to control access to the functions provided in the smart contract. Each user is uniquely identified by their public key, and this key can be used to determine their access to the functions on the smart contract.
- **Availability:** Availability is a feature that is provided by the blockchain itself because of its decentralized nature. Every node in the network has their own copy of the ledger, so even if one node becomes unavailable, the whole system would still be accessible. This removes a single point of failure and introduces fault tolerance in the system.

4. Results

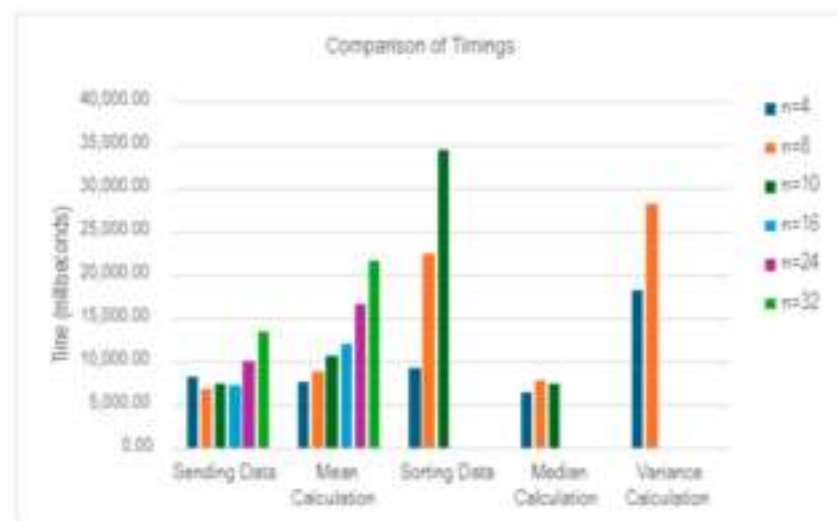
We ran the proof-of-concept implementation of the system, where the smart contract has the functionality to calculate the mean, median, and variance of the data set. We recorded the time required to perform various operations for data points of 4, 8, 10, 16, 24, and 32. These operations include setting the data (i.e., sending the data to the smart contract), mean calculation, sorting the data (which is necessary for finding the median), median calculation, and variance calculation. All experiments were carried out with a 16-bit data type. Each data set was randomly generated with points in the range of 0 to 50. Each produced data set was sent to the smart contract at once. Each data set size was tested eight times to ensure the reliability of the results. The tests were performed at different times and at different geographical locations with different connectivity. Table 1 shows the average timing of eight trials for each data set size.

Table 1. Timing of operations from request to result in milliseconds.

Number of Data Points (n)	Sending Data	Mean Calculation	Sorting Data	Median Calculation	Variance Calculation
4	8088.66679	7512.111693	9063.634214	6312.926094	18,109.45269
8	6642.155596	8792.629253	22,330.79229	7679.880587	28,161.12535
10	7243.960528	10,486.1769	34,412.11508	7358.729393	
16	7111.459872	11,948.44701			
24	9990.13181	16,596.62563			
32	13,358.13821	21,479.25259			

As exhibited in Table 1, the sizes of some data sets were not suitable for certain operations due to the gas limit imposed by Zama. For example, while the sorting process was performed on data sets containing 4, 8, and 10 data points, the variance process was applied only to data sets containing 4 and 8 data points. Sending the data sets to the blockchain and calculating the mean value could be performed on all data sets.

Figure 3 compares the timings for each operation based on the number of data points. It also compares the timings for different operations. Overall, as the number of data points increases, the time required for an operation also increases. A few exceptions to this are sending data for a smaller number of data points and the median calculation. We attribute this to the fact that operations that require a short time to complete are more susceptible to being impacted by the volume of transactions as well as the quality of the network connectivity. As depicted in the graph, variance and sorting take more time than setting data, mean calculation, and median calculation. Median calculation takes the least time since it involves a few simple operations compared to the other operations regardless of the data size.

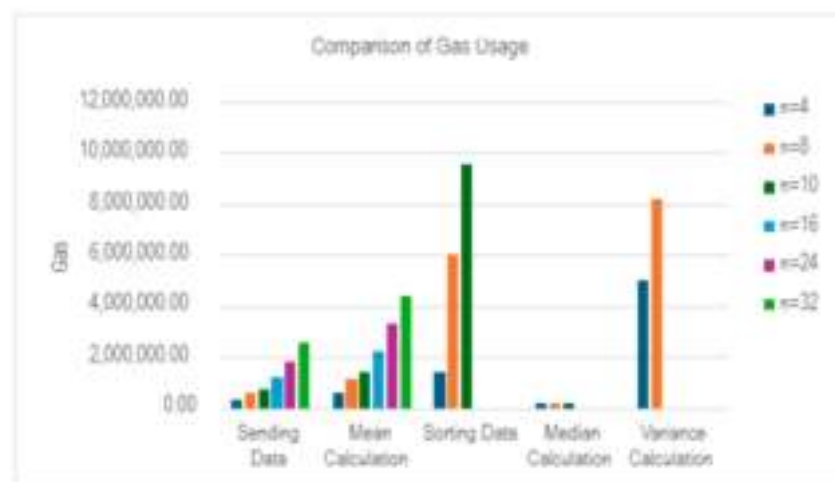
**Figure 3.** Comparison of timings of operations with respect to number of data points.

We also recorded the gas usage of each operation for each test and verified that the data sort and variance calculation reached the gas limit imposed by Zama at 10 data points and 8 data points, respectively. Table 2 shows the averages of the eight trials for each data set size.

Table 2. Gas usage of operations.

Number of Data Points (n)	Sending Data	Mean Calculation	Sorting Data	Median Calculation	Variance Calculation
4	290,491.5	577,618	1,367,854	169,517	5,022,035
8	565,458.875	1,124,706	6,007,343	169,517	8,213,723
10	709,171.5	1,398,250	9,565,005	169,517	
16	1,166,246.625	2,218,883			
24	1,835,907.625	3,313,060			
32	2,574,607.375	4,407,237			

It is no wonder that the gas usage was consistent for the given number of data points for the same operation. Figure 4 shows a comparison of the gas usage relative to the number of data points for each operation as well as a comparison of the gas usage with respect to the different operations.

**Figure 4.** Comparison of gas usage of operations with respect to number of data points.

As Figure 4 depicts, gas usage increased with the number of data points except for the median. The median calculation involved a simple operation of finding the index of the middle element, requiring only one addition and one division regardless of the number of data points. Hence, gas usage was the same for all three data set sizes. Calculating the median requires data to be sorted. However, because sorting data requires too much gas, it cannot be calculated for more than 10 data points. Similarly, variance calculation required around 10 million gas for eight data points. Adding an additional data point would require more than 10 million gas, which would be over the gas limit imposed by Zama. As is evident from the graph and the description of the implementation of the operations in Section 3, the operations that require more computations require more gas.

5. Discussion

This study proposed a system that provides data security, statistical privacy, and confidentiality using techniques and tools purely from mathematics and computer science. The system utilizes fully homomorphic encryption over a blockchain and ensures end-to-end security of the data from the moment they leave the data owner until the researcher receives the computation results. The data are secured in transit, at rest, as well as in use. Furthermore, the proposed system also provides computational integrity. With all the protections it provides, the system is an ultimate solution for meeting data security requirements.

A proof-of-concept implementation for the proposed system is carried out on the public blockchain Zama. In the implementation, a researcher can request secure calculations of mean, median, and variance. The timing results for carrying out the required operations on various numbers of data points are presented, along with the limitations of working with homomorphic encryption libraries and a public blockchain. While this study successfully demonstrates the possibility of securing data and computations on a blockchain using fully homomorphic encryption, there are certain challenges and areas that need improvement.

Fully homomorphic operations are quite expensive in terms of resources, and there are limitations on the number of calculations that can be performed on the blockchain before exceeding the gas and cost limits. This is especially critical when running the application on the main network, where transactions incur fees. This limitation can be partially mitigated on a private or permissioned blockchain network. Such a blockchain provides similar guarantees as the public blockchain and gives more control over who can use the application.

Future studies in this area can incorporate more complex algorithms across various disciplines, ranging from statistics to machine learning, and include a wider data range as more resources become available and technology advances. Furthermore, traditional algorithms that have been designed for plaintext data, such as searching and sorting, need to be carefully revised to support homomorphic computations, subject to the capabilities of the libraries available. Another possible direction for future studies could be to develop a framework that supports fully homomorphic computations over that smart contract and provides a wider range of functionality while overcoming the limitations of the current system, such as allowing division of two encrypted numbers and incorporating decimal or floating point numbers.

Author Contributions: Conceptualization, Y.K.P.; Data curation, R.R., Y.K.P. and Z.D.M.; Formal analysis, Y.K.P. and Z.D.M.; Investigation, R.R., Y.K.P. and Z.D.M.; Methodology, R.R. and Y.K.P.; Project administration, Y.K.P.; Resources, R.R., Y.K.P. and Z.D.M.; Software, R.R. and Z.D.M.; Supervision, Y.K.P.; Validation, R.R., Y.K.P. and Z.D.M.; Visualization, Y.K.P. and Z.D.M.; Writing—original draft, R.R. and Y.K.P.; Writing—review and editing, R.R., Y.K.P. and Z.D.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article/ Further inquiries can be directed to the corresponding author(s).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dwork, C.; Smith, A.; Steinke, T.; Ullman, J. Exposed! A Survey of Attacks on Private Data. *Annu. Rev. Stat. Appl.* **2017**, *4*, 61–84. [\[CrossRef\]](#)
2. Kitamura, K.; Irvan, M.; Yamaguchi, R.S. Anonymity test attacks and vulnerability indicators for the ‘Patient characteristics’ disclosure in medical articles. In Proceedings of the 2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Genoa, Italy, 6–10 June 2022; pp. 186–193. [\[CrossRef\]](#)
3. Nissim, K. Privacy: From Database Reconstruction to Legal Theorems. In Proceedings of the 40th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, in PODS’21, Virtual Event, China, 20–25 June 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 33–41. [\[CrossRef\]](#)
4. Wang, Y.-R.; Tsai, Y.-C. The Protection of Data Sharing for Privacy in Financial Vision. *Appl. Sci.* **2022**, *12*, 7408. [\[CrossRef\]](#)
5. Agarwal, S. Sray and Mishra Data and Model Privacy. In *Responsible AI: Implementing Ethical and Unbiased Algorithms*; Springer International Publishing: Cham, Switzerland, 2021; pp. 153–170. [\[CrossRef\]](#)
6. Sokolovska, A.; Kocarev, L. Integrating Technical and Legal Concepts of Privacy. *IEEE Access* **2018**, *6*, 26543–26557. [\[CrossRef\]](#)
7. Kifer, D.; Lin, B.-R. An Axiomatic View of Statistical Privacy and Utility. *J. Priv. Confidentiality* **2012**, *4*. [\[CrossRef\]](#)
8. Slavković, A.; Seeman, J. Statistical Data Privacy: A Song of Privacy and Utility. *Annu. Rev. Stat. Appl.* **2023**, *10*, 189–218. [\[CrossRef\]](#)
9. C.3. Statistical Confidentiality—MSITS 2010 Compilers Guide—UN Statistics Wiki. Available online: <https://unstats.un.org/wiki/display/M2CG/C.3.++Statistical+confidentiality> (accessed on 4 June 2024).

10. UNSD—Fundamental Principles of National Official Statistics. Available online: <https://unstats.un.org/fpos/> (accessed on 5 June 2024).
11. Statistical Confidentiality and Personal Data Protection—Eurostat. Available online: <https://ec.europa.eu/eurostat/web/microdata/statistical-confidentiality-and-personal-data-protection> (accessed on 18 June 2024).
12. National Academies of Sciences, Engineering, and Medicine; Division of Behavioral and Social Sciences and Education; Committee on National Statistics; Panel on Improving Federal Statistics for Policy and Social Science Research Using Multiple Data Sources and State-of-the-Art Estimation Methods; Groves, R.M.; Harris-Kojetin, B.A. (Eds.) *Innovations In Federal Statistics: Combining Data Sources While Protecting Privacy*; The National Academies Press: Washington, DC, USA, 2017. [CrossRef]
13. Kumar, R.; Kumar, J.; Khan, A.A.; Ali, H.; Bernard, C.M.; Khan, R.U.; Zeng, S. Blockchain and homomorphic encryption based privacy-preserving model aggregation for medical images. *Comput. Med. Imaging Graph.* **2022**, *102*, 102139. [CrossRef] [PubMed]
14. Liang, W.; Zhang, D.; Lei, X.; Tang, M.; Li, K.-C.; Zomaya, A. Circuit Copyright Blockchain: Blockchain-Based Homomorphic Encryption for IP Circuit Protection. *IEEE Trans. Emerg. Top. Comput.* **2020**, *9*, 1410–1420. [CrossRef]
15. Yaji, S.; Bangera, K.; Neelima, B. Privacy Preserving in Blockchain Based on Partial Homomorphic Encryption System for Ai Applications. In Proceedings of the 2018 IEEE 25th International Conference on High Performance Computing Workshops (HiPCW), Bengaluru, India, 17–20 December 2018; pp. 81–85. [CrossRef]
16. Mutlu, Z.D.; Peker, Y.K.; Aydın, A. Selçuk Blockchain-based Privacy Preserving Linear Regression. *J. Millimeterwave Commun. Optim. Model.* **2023**, *3*, 45–49. Available online: <https://www.jomcom.org/index.php/1/article/view/81> (accessed on 5 June 2024).
17. Vanin, F.N.; Policarpo, L.M.; Righi, R.D.; Heck, S.M.; da Silva, V.F.; Goldim, J.; da Costa, C.A. A Blockchain-Based End-to-End Data Protection Model for Personal Health Records Sharing: A Fully Homomorphic Encryption Approach. *Sensors* **2023**, *23*, 14. [CrossRef] [PubMed]
18. Umar, B.; Olaniyi, O.; Olajide, D.; Dogo, E. Paillier Cryptosystem Based ChainNode for Secure Electronic Voting. *Front. Blockchain* **2022**, *5*, 927013. [CrossRef]
19. Shrestha, R.; Kim, S. Integration of IoT with blockchain and homomorphic encryption: Challenging issues and opportunities. *Adv. Comput.* **2019**, *115*, 293–331. [CrossRef]
20. Caldarola, F.; d’Atri, G.; Zanardo, E. Neural Fairness Blockchain Protocol Using an Elliptic Curves Lottery. *Mathematics* **2022**, *10*, 3040. [CrossRef]
21. fhevm/fhevm-whitepaper.pdf at main·zama-ai/fhevm·GitHub. Available online: <https://github.com/zama-ai/fhevm/blob/main/fhevm-whitepaper.pdf> (accessed on 4 June 2024).
22. Estimate Gas | 0.3 | fhEVM. Available online: <https://docs.zama.ai/fhevm/v/0.3-2/how-to/gas> (accessed on 4 June 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Modelling Analysis of Channel Assembling in CRNs Based on Priority Scheduling Strategy with Reserved Queue

Qianyu Xu ¹, Suoping Li ^{2,*}, Jaafar Gaber ³  and Yuzhou Han ²

¹ School of Electrical & Electronic Engineering, Xi'an Aeronautical University, Xi'an 710077, China; 202202001@xaau.edu.cn

² School of Electrical & Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China; 19119376252@163.com

³ Department of Computer Science and Computer Engineering, Universite de Technologie Belfort-Montbéliard, 90010 Belfort, France; gaber@utbm.fr

* Correspondence: lsuop@lut.edu.cn

Abstract: In cognitive radio networks, channel assembling allows secondary users (SUs) to expand network capacity and improve spectrum utilization. Scheduling strategies only based on heterogeneous service classification cannot guarantee the delivery priority of vital elastic services in special scenarios such as emergency rescue. Therefore, a priority scheduling strategy with reserved queue (Ps-rq) is proposed in this work. A static factor is defined to classify SUs into elastic services and real-time services based on message type, while a dynamic factor is defined to differentiate high-priority elastic services based on information validity, message correlation and message size. The high-priority users in the interrupted elastic services are placed in the reserved queue to ensure its services. Accordingly, the scheduling algorithm and the dynamic channel access process is presented. A continuous-time Markov chain analysis is conducted and all possible transition states, trigger events, transition rates and transition conditions of the system starting from a general state are derived. Furthermore, evaluation indexes of system performance are obtained. Study cases and simulation results prove that the proposed strategy can enhance network capacity, reduce blocking probability and forced termination probability for secondary users, and notably enhance the performance of high-priority elastic services. In addition, we analyze the characteristics of Ps-rq through a comprehensive comparison with four other schemes. The experiment proves that the Ps-rq strategy can effectively improve the service quality of the vital elastic services on the basis of providing fair scheduling.

Keywords: cognitive radio networks; channel assembling; priority scheduling strategy; reserved queue; Markov chain; spectrum utilization; transition conditions



Citation: Xu, Q.; Li, S.; Gaber, J.; Han, Y. Modelling Analysis of Channel Assembling in CRNs Based on Priority Scheduling Strategy with Reserved Queue. *Electronics* **2024**, *13*, 3051. <https://doi.org/10.3390/electronics13153051>

Academic Editors: Duc Tai Le and Syed Muhammad Raza

Received: 11 June 2024

Revised: 29 July 2024

Accepted: 30 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Traditional wireless communication networks are mainly based on static spectrum allocation. However, the licensed user always occupies a fixed frequency band, which will lead to an excessive waste of spectrum [1]. Cognitive radio networks (CRNs) allow secondary users (SUs) to explore and use the unused spectrum of primary users (PUs) through spectrum sensing to improve spectrum utilization [2,3].

Improving spectrum utilization requires cognitive radio systems to flexibly schedule and allocate available spectrum resources [4,5]. Dynamic spectrum access (DSA) can ensure SUs get access to spectrum gaps in the licensed spectrum bandwidth [6]. The flexibility of DSA is reflected in that the ongoing SUs can exit the channel once sensing the signals of the PUs, then spectrum adaptation can be performed through spectrum handoff or dynamic channel selection. Designing resource scheduling rules based on DSA that conform to the application environment can help improve the performance of the secondary network [7].

In CRNs, the adoption of channel assembling technology and spectrum adaptation technology in the process of DSA can significantly enhance the service efficiency of SUs. Channel assembling technology allows SUs to bind or aggregate one idle channel to more to increase the secondary network capacity. Combined with spectrum adaptation technology, the ongoing SUs can adaptively adjust the number of assembled channels according to the activities of other PUs and SUs to provide flexible scheduling. However, the modeling of the DSA process based on resource allocation is complex, so it is necessary to analyze and evaluate the performance of secondary networks to help design operation rules.

The service performance of SUs is inseparable from the resource scheduling and allocation method. The service flow in wireless communication can be divided into real-time service (such as audio and video) and elastic service (such as data) based on delay sensitivity. Real-time service usually has higher priority than elastic service, while elastic service has lower delay sensitivity, and its transmission rate can be adjusted adaptively according to the network environment.

Researchers proposed a variety of resource allocation methods for heterogeneous traffic to improve spectrum utilization [8–14]. The main technique is to aggregate the available spectrum resources into the spectrum pooling according to the dynamic network environment and then allocate them on demand. Among them, refs. [10–13] studied the resource scheduling problem of multi-channel cognitive radio networks with heterogeneous traffic based on channel assembling and used mathematical methods to establish the analysis model to evaluate the performance of secondary networks. Previous research [11] proved that arranging a variable number of assembled channels for elastic services and a fixed number of assembled channels for real-time services according to the operating environment can help improve spectrum efficiency.

In [15], the authors proposed that the adoption of the continuous-time Markov chain (CTMC) process can provide a modeling method for the performance analysis of CRNs with channel assembling, which has the advantage of avoiding the time synchronization problem of PUs and SUs on the same channel. In [11], the authors proposed a channel assembly strategy considering spectrum adaptation and heterogeneous services, and CTMC was used to evaluate the secondary network performance. However, the SU interrupted by the PU cannot continue the service after exiting the channel, which obviously increases the possibility of the SU being dropped. Therefore, ref. [12] proposed to establish queues for heterogeneous services and adopt a queue scheduling algorithm to improve network capacity on the basis of prioritization. Previous research [12,16] proved that establishing queues for SUs can improve network capacity and reduce the blocking probability of SUs. By considering a variety of factors that affect channel capacity, including a time-varying wireless link, a finite buffer, traffic classes and an adaptive modulation and coding (AMC) scheme, a channel assembling strategy with queues was proposed in [17] and the interaction between system parameters was investigated by establishing a CTMC analysis model. Based on channel assembling, an adaptive spectrum leasing algorithm was proposed in [18], and the DSA performance of SU traffic with access priority was discussed by developing the CTMC model. The above research shows that the adoption of channel assembling technology for heterogeneous services can improve the utilization of idle spectrum. In addition, the establishment of CTMC can help to model the scheduling dynamics of spectrum resources among users in the resource allocation problem. The aforementioned research presents a more rigorous and refined methodology for mathematically analyzing the performance of the secondary system.

In CRNs with multi-type services, resource allocation based on service classification can improve the transmission reliability of important information. However, the improvements usually result in the starvation problem of low-priority services due to a lack of service for a long time. Resource scheduling schemes that can alleviate the starvation problem of low-priority users were proposed in [19–21], and the secondary system performance was evaluated by establishing Markov analysis models with queues. In [20], channel binding technology with starvation mitigation was adopted, and a non-pre-emptive priority

M/G/1 queuing model was used to depict the spectrum handoff process of SUs, so as to improve the resource utilization. A round robin priority (RRP) scheduling algorithm combining static and dynamic resource allocation was proposed in [21]. This algorithm obtained the transmission opportunities of low-priority users by temporarily suspending the high-priority queue. The purpose of this is to minimize the starvation of low-priority services in CRNs. In fact, the RRP algorithm in [21] adopts queue polling scheduling, but when the traffic of one queue is large, user scheduling only based on queue polling will greatly reduce the system performance [22]. The scheduling of heterogeneous services, based on service type division, can indeed offer a more convenient priority scheduling scheme for channel allocation and priority scheduling in CRNs during system initialization. The previous research, however, failed to address the transmission requirements of high-priority elastic users in practical applications, thus failing to provide a fundamental solution to the issue of service starvation caused by scheduling priorities.

Consequently, this motivates the investigation of our study. This paper investigates resource scheduling and allocation with service classification for CRNs that require ranking services in special scenarios. Some important elastic services need to be given higher priority in special CRN application scenarios, including emergency rescue, drones [23] and UAVs. Therefore, considering that scheduling schemes with hierarchical services often lead to the starvation problem of low-priority services due to long periods of unserved services, the goal of this study is to improve the delivery efficiency of important information based on providing as fair a distribution mechanism as possible.

Different from our previous studies [9], this paper considers a more general scheduling scheme based on heterogeneous services. The aim of this study is to propose a more comprehensive resource scheduling and allocation scheme for CRNs, which can guarantee the QoS of high-priority elastic services. More specifically, based on the polling scheduling, a new reserved queue is provided for the elastic services with high-priority that are interrupted by PUs, and the highest priority is arranged for the SUs in the reserved queue. Based on channel assembling and spectrum adaptation technology, the dynamic spectrum access process of heterogeneous services is analyzed and spectrum resource allocation among SUs is investigated through developing a CTMC. The major contributions of this study are as follows:

1. This study proposes a priority scheduling strategy with reserved queue (Ps-rq strategy). The system model and service classification scheme of the proposed strategy are presented. By combining channel assembling and spectrum adaptation technology, a proportional priority allocation scheme based on Ps-rq strategy is given.
2. The resource flow process of the secondary network is mapped on a CTMC. By employing stochastic analysis methods, the complex user activities in the CRN are associated with state transitions individually. The transition event, transition rate and transition condition that can trigger changes in the system states are provided, thereby offering a theoretical foundation for enhancing the performance of secondary systems through channel assembling.
3. The procedure for dimension reduction of high-dimensional Markov chains is given. Based on this, the analytical results of the secondary network performance are investigated and numerically simulated. We present a comparison of the Ps-rq with four previous schemes, and it is proven through numerical verification that the proposed strategy ensures service quality for interrupted important elastic services while maintaining fair scheduling.

2. Priority Scheduling Strategy with Reserved Queue (Ps-rq Strategy)

This section provides a heterogeneous service classification scheme for CRNs with multi-type services. On this basis, a priority scheduling strategy with reserved queue (Ps-rq) is proposed. The purpose is to provide a fair scheduling for CRNs with multi-type services, which can guarantee the delivery reliability of important information while alleviating the starvation problem of low-priority queues.

2.1. System Model and Assumptions

Centralized cognitive radio networks (CRNs) consist of two types of radios, primary user (PUs) and secondary user (SUs). User scheduling is controlled by the corresponding base station. The central node is responsible for interacting with the surrounding environment, as well as managing spectrum allocation and scheduling decisions to other SUs through a control channel. The central node in centralized CRNs can enhance system performance by making efficient decisions and reducing interference among users. In contrast to distributed CRNs, centralized CRNs eliminate the need for further deliberation on aspects such as inter-user interference and node selfishness.

We assume that there are M ($M \in \mathbb{Z}^*$) PUs sharing the spectrum bandwidth, and each PU can access only one channel. Compared with SUs, PUs have absolute channel access priority. SUs are equipped with the means for spectral energy detection, assuming that SUs have perfect sensing ability. And the sensing time required is so short that PUs will not reach the channel within this period. Once the PU is sensed, the ongoing SU will immediately release the assembled channels and exit and perform spectrum adaptation at the same time. Assuming that there is less time spent on spectrum adaptation, the channel switching will not affect the transmission of SUs.

2.2. Heterogeneous Service Classification

Supported by the system model, this section proposes a heterogeneous service classification scheme to satisfy the scheduling demands for CRNs with multi-type services including data, image, audio and video. For instance, delayed delivery or packet loss of vital information of patients during a medical emergency may be life threatening, and this part of the information is mainly an elastic service. However, a previous study [12] based on heterogeneous service classification did not consider important elastic services with high delivery priority under special scenarios. The transmission priority and reliability of important elastic services can hardly be ensured. On the basis of heterogeneous service classification, multi-type services in CRNs are also classified by dynamic factors, and the elastic services with high-priority are separated. The details are as follows:

- Static factor: message type. SUs in CRNs are classified into elastic services (SU_e) and real-time services (SU_r) according to delay sensitivity. Elastic services have the characteristics of short frame length, heavy traffic flow and high delay tolerance. Real-time services require the timeliness of data transmission. According to the tolerance of delay, the service priority is defined as follows:

$$\text{Priority} : SU_r > SU_e;$$

- Dynamic factor: elastic services in CRNs are reclassified based on three dynamic factors including information validity, message correlation and message size.

Information validity (Val) refers to the time remaining before the deadline. Messages with shorter times remaining have higher priority.

$$Val = \frac{\text{The time remaining before the deadline}}{\text{The delivery time of information}}; \quad (1)$$

Message correlation (Cor) refers to the type of message that needs to be delivered first.

$$\begin{cases} Cor = 1, & \text{Relevance of important information;} \\ Cor = 0, & \text{Others} \end{cases}; \quad (2)$$

Message size ($Size$) is inversely proportional to the message priority.

$$Size = \frac{\text{The message size of the SU}}{\text{Average message length}}; \quad (3)$$

After combining the above three dynamic factors, elastic services are classified into high, medium and low priorities. The purpose is to identify the elastic services with high-priority (SU_e^h), and then prioritize them over real-time services. The service priority is suggested as follows:

$$\text{Priority} : SU_e^h > SU_r > SU_e$$

Furthermore, a three-dimensional (3D) priority view of elastic services is presented in Figure 1. Takagi–Sugeno fuzzy control is used to prioritize elastic services [24]. The input value of the fuzzy inference system ranges from 0 to 1. The output of the system is priority and is divided into three fuzzy sets, whose domains are $[-1, 1]$. According to the fitness, the “If-then” rule is adopted to set 27 fuzzy rules, and the priority of elastic services are divided into high, medium and low priorities. For example, if $Val = 0$, $Cor = 1$ and $Size = 1$, then $Priority = 1$, which represents important information with a short remaining time and a small message size. From the input and output variables, Gaussian and trapezoidal membership functions are used to obtain the 3D priority view of elastic services.

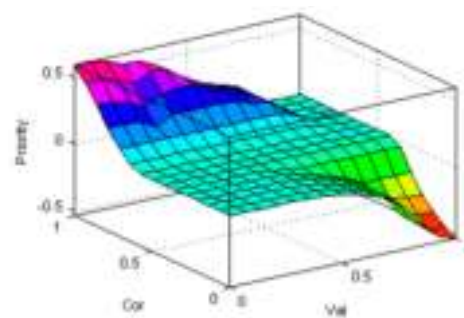


Figure 1. 3D priority view of elastic services.

2.3. Ps-rq Strategy

Based on the above heterogeneous service classification scheme, traffic with multi-type services entering the system is served by the priority scheduling strategy with reserved queue (Ps-rq strategy). As shown in Figure 2, the Ps-rq strategy consists of an information scheduling unit and a data transmission unit. Users entering the secondary system (SUs) are divided into the elastic service (SU_e) and real-time service (SU_r) according to a static factor through a packet classifier I. (Packet classifiers are employed for the purpose of parsing incoming packet types.) Different physical queues are arranged for SU_e and SU_r , which are denoted L_e and L_r . If the service of SUs is interrupted by a PU during the transmission, spectrum adaptation will be performed. Considering that SU_r requires high timeliness and has jitter tolerance, if the spectrum adaptation fails, the interrupted SU_r will be dropped and release the assembled channels, and it will not return back to the end of L_r . That is, L_r has no feedback loop.

Different from SU_r , SU_e has low requirements on timeliness and may contain vital signals. Thus, a feedback loop is arranged for L_e . The interrupted SU_e will re-entered into packet classifier II and classified into high, medium and low priorities based on a dynamic factor. The purpose is to separate high-priority elastic services, mark them as SU_e^h , and put them into the reserved queue L_e^h . Meanwhile, the interrupted medium-priority elastic service (SU_e^m) and low-priority elastic service (SU_e^l) are removed from their marks and returned to the back of the queue L_e (feedback loop). Assuming that each queue is isomorphic, the secondary users in the queue are waiting to be scheduled according to the order of first come first service (FCFS). The priority of the queue is set to $L_e^h > L_r > L_e$. SUs of different types can only line up in the corresponding queue. The service of SUs will be blocked if there is no space left in the queue. If it is full when the interrupted SU_e and SU_e^h enter the corresponding queue, the service will be forced to terminate.

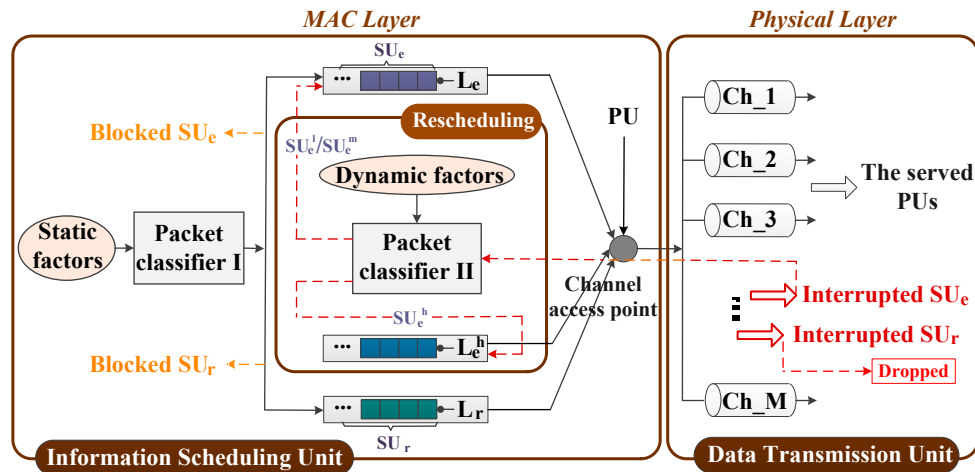


Figure 2. Ps-rq strategy.

2.4. Channel Allocation Based on Ps-rq Strategy

In the scheduling process, channel assembling technology is adopted to improve the service rate of the SUs by assembling one channel or more to SUs. Meanwhile, spectrum adaptation technology is used to allow adaptive allocation of the assembled channels. For CRNs with multi-type services, channel allocation based on the Ps-rq strategy is as follows:

The SU_r and the marked SU_e^h are assembled with a fixed number of channels to ensure the transmission efficiency. More specifically, the service of SU_r can be started by aggregating a channels, where $a \in \mathbb{Z}^*$. Accordingly, SU_e^h can start the service by aggregating b channels, where $b \in \mathbb{Z}^*$. The difference is SU_e can flexibly adjust the number of assembled channels based on the spectrum adaptation technology. To be specific, the service of SU_e can be started by aggregating w channels, and the service rate can be improved by increasing the aggregated channels to the upper bound v , where $w, v \in \mathbb{Z}^*$ and $w \leq v$. For the convenience of starting the service of SU_r and the marked SU_e^h , the parameter setting of this study is based on $a = b < w$. Suppose that the service rate of SUs increases linearly after the channel assembling technology is adopted, the service rate of SU_r with a assembling channels increases by a times.

Channels are allocated to SUs based on proportional channel allocation with priority. According to the queue priority $L_e^h > L_r > L_e$, if there are N_{free} idle channels, detect whether the three queues are empty. If none of them are empty, the idle channel is allocated to SU_e^h , SU_r and SU_e in the queue in proportion of $1 : \alpha : \beta$, where $\alpha, \beta \geq 1$. Here, the value of α, β can be determined according to the priority requirements and delay tolerance of SU_e^h , SU_r and SU_e . Then, the remaining channels are allocated to SU_e^h according to priority, followed by SU_r and SU_r . If only two of the queues are not empty, the allocation process is the same as above. If only one is not empty, all N_{free} channels are allocated to this queue. Furthermore, if there are still idle channels, the remaining channels are allocated to the ongoing SU_e . The above allocation process is realized in MATLAB, and the proportional channel allocation with priority is shown in Scheme 1. Here, the number of idle channels is still determined by SUs obtaining available idle channel information through spectrum energy detection technology to achieve opportunistic access.

```

1. Start ..... % Step 1
2. Determine the number of the current idle channels  $N_{free}$  ..... % Step 2
3. Determine the current queue length of  $L_e$ ,  $L_r$  and  $L_e^h$  ..... % Step 3
4. If None of three queues is idle ..... %  $SU_e: SU_r: SU_e^h = 1: \alpha: \beta$  % Step 4
5.   Then calculate allocation factor:  $s = \lfloor N_{free} / (1 + \alpha + \beta) \rfloor$ 
6.       calculate the allocated channels for  $SU_e, SU_r, SU_e^h$ :  $n_e = s, n_r = \alpha \cdot s, n_e^h = \beta \cdot s$ 
7.       calculate current remaining channels:  $t = \lfloor N_{free} - (1 + \alpha + \beta) \rfloor + \lfloor n_e \bmod w \rfloor + \lfloor n_r \bmod a \rfloor + \lfloor n_e^h \bmod b \rfloor$ 
8.   Repeat Step 4 ..... % Step 5
9.   Until  $s = 0$ 
10. End loop
11. If  $N_{free} > 0$  ..... % Step 6
12.   Then determine the remaining queue length of  $L_e^h, q_e^h$  ..... % Priority  $SU_e^h > SU_r > SU_e$ 
13.     If  $q_e^h \cdot b > t$  Then allocate  $\lfloor \frac{t}{b} \rfloor \cdot b$  channels to  $SU_e^h$ 
14.     Else allocate  $q_e^h \cdot b$  channels to  $SU_e^h$ 
15.     End if
16.   Then determine the remaining queue length of  $L_r, q_r$  ..... % Step 7
17.     Repeat Step 6
18.     Until  $s = 0$ 
19.     End loop
20.   Then determine the remaining queue length of  $L_e, q_e$  ..... % Step 8
21.     Repeat Step 6
22.     Until  $s = 0$ 
23.     End loop
24.   If there still exist unallocated channels
25.   Then determine the current queue length of  $L_e^h, L_r$  and  $L_e$  ..... % Step 9
26.     If at least one queue is not idle
27.     Then the unallocated channels will wait in the system % Waiting for binding with the new channel
28.     Else allocate the unallocated channels to the ongoing  $SU_e$  % Increase the service rate of  $SU_e$ 
29.     End if
30.   End if
31. Else
32. End if
33. If  $L_r > 0, L_e^h > 0$  Repeat Step 4-9 ..... %  $SU_r: SU_e^h = \alpha: \beta$ 
34. If  $L_e > 0, L_e^h > 0$  Repeat Step 4-9 ..... %  $SU_e: SU_e^h = 1: \beta$ 
35. If  $L_e > 0, L_r > 0$  Repeat Step 4-9 ..... %  $SU_e: SU_r = 1: \alpha$ 
36. If only  $L_e^h > 0$  Then allocate  $\lfloor \frac{N_{free}}{b} \rfloor$  channels to  $SU_e^h$ 
37. If only  $L_r > 0$  Then allocate  $\lfloor \frac{N_{free}}{a} \rfloor$  channels to  $SU_r$ 
38. If only  $L_e > 0$  Then allocate  $\lfloor \frac{N_{free}}{w} \rfloor$  channels to  $SU_e$ 
39. If still remain idle channels Then allocate the idle channels to ongoing  $SU_e$ 
40. End if

```

Scheme 1. Proportional channel allocation with priority.

3. Dynamic Channel Access Process

In CRNs, the arrival and departure behaviors of PUs and SUs result in a change in the system state. In order to analyze the resource flow process of the SUs in different system states, the dynamic channel access process based on the proposed Ps-rq strategy is investigated in this section through classifying the user activities in CRNs into four events: PU arrival, PU departure, SU arrival and SU departure.

- Event A: PU arrival

SUs in CRNs are classified into elastic services and real-time services according to delay sensitivity. Elastic services have the characteristics of short frame length, heavy traffic flow and high delay tolerance. Real-time services require the timeliness of data transmission. According to the tolerance of delay, the service priority is defined as follows:

When the PU arrives, if there is an idle channel in the system, it will be allocated to the newly arrived PU. In this case, the services of other SUs will not be affected. But if there are no idle channels in the system, the PU will interrupt any one ongoing channel of the SUs due to the higher data access priority. The interrupted SU will exit the channel and perform spectrum adaptation. That is, the ongoing SU_e with the maximum number of channels will donate the channel to the interrupted SU until the assembled channels of SU_e cannot support the service, then the ongoing SU_e with the second maximum number of channels will donate its channel ... until all of the SU_e cannot provide the required number of channels of the interrupted SU. Then, the adaptation fails, which means that all

the ongoing SU_e cannot donate the channel for the interrupted SU. The SU_e who fails the adaptation will enter packet classifier II and mark the priority through a fuzzy controller. If the SU_e belongs to SU_e^h , it will enter the reserved queue L_e^h ; otherwise, it will return back to the end of queue L_e . If the queue is full, the service of the SU will be forced to terminate.

If the PU pre-empts the channel of a SU_r or a marked SU_e^h , spectrum adaptation will be performed in the same way. The service will continue when the adaptation is successful; otherwise, the interrupted SU_r or SU_e^h will be forced to terminate and release the assembled channel.

In particular, the channels released by the forcibly terminated SU_e^h or SU_r cannot be used to start the service of SUs in the queue due to the setting of $a = b < w$. However, the channels released by the forcibly terminated SU_e can be used for the queuing SU_e^h and SU_r in L_e^h and L_r . The channel allocation of SUs in the queue follows Scheme 1. If no SU in the queue uses the idle channel, these channels will be assembled to other SUs, obeying Rule 1.

Rule 1: The idle channel will be preferentially assembled to the ongoing SU_e with the minimum aggregated channels until the SU_e is assembled to the upper bound, v . If there are still idle channels, the remaining channels will be assembled to the ongoing SU_e with the second minimum aggregated channels until all of the ongoing SU_e reach the upper bound, v .

We present a flow chart, as shown in Figure 3, to show the dynamic channel access process based on the Ps-rq strategy on PU arrival. Similarly, other events including PU departure, SU arrival and SU departure can also use the flow chart to describe the channel access process.

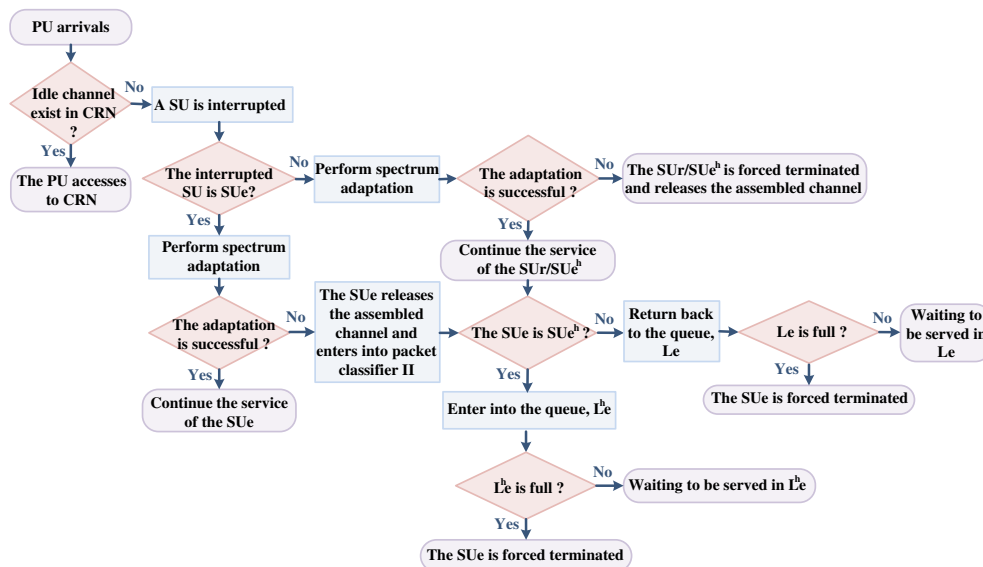


Figure 3. The dynamic channel access process based on the Ps-rq strategy on PU arrival.

- **Event B: PU departure**

The departure of a PU releases a channel. The released channel will be preferentially used for the SUs waiting in the queue. If the SUs in the queue do not use the idle channel, the idle channel can be used for the ongoing SU_e whose assembled channels have not reach the upper bound with the channel assembling technology.

Specifically, the idle channel is preferentially allocated to SU_e^h in L_e^h , obeying the rule of first come first service (FCFS). The service of SU_e^h can be started only when $b - 1$ free channels exist in the system. If the SU_e^h in the queue do not use the idle channel, the channel can be assembled to SU_r in L_r , and the service of SU_r can be started only when $a - 1$ free channels exist in the system. If both SU_e^h and SU_r in the queue do not use the idle channel, the channel can be assembled to SU_e in L_e . Similarly, the service of SU_e can be started only

when $w - 1$ idle channels already exist in the system. If no SU in the queue uses the idle channel, the channel will be assembled to the ongoing SUs in the system, obeying Rule 1.

- Event C: SU arrival

There are three kinds of SU arrivals in the system including SU_e^h , SU_e and SU_r . If the SU arriving at the system is SU_e^h and there are b idle channels in the system, then the service of the newly arrived SU_e^h can be started. Similarly, the service of the newly arrived SU_e or SU_r can be started up when the number of channels existing in the system reaches the minimum value w or a . If the free channels in the system cannot support the service, the ongoing SU_e will perform spectrum adaptation to donate the channel for the newly arrived SUs. If all of the ongoing SU_e are unable to provide the required number of channels, the newly arrived SU will be ranked at the end of the corresponding queue. If the queue is full, the service of the SU will be blocked. However, since all the SU_e^h entering the queue L_e^h come from the interrupted SU_e , when L_e^h is full, the service of SU_e^h will be forced to terminate, that is, there is no blocked SU_e^h .

- Event D: SU departure

The departure of SU releases all aggregated channels. That is, the departure of SU_e releases k ($w \leq k \leq v$) channels; the departure of SU_r releases a channels; and the departure of SU_e^h releases b channels. Scheme 1 is used to allocate the channels released by SUs, where N_{free} is the sum of the number of channels released by SUs and the number of the idle channels in the system. If there are still idle channels after all SUs waiting in the queue have been assembled, the idle channel will be allocated to the ongoing SU_e , obeying Rule 1.

4. CTMC Analysis and QoS Measures

In the channel access process, user activities trigger the change in system state. Continuous-time Markov chains (CTMCs) are used to set up a corresponding relationship between the possible state transition of the system and the triggering events that cause the change in the system state, which can depict the spectrum resource flow among multi-type SUs. Therefore, the performance analysis of the secondary system can be carried out by means of the CTMC. The activities of the SUs can be mapped in the multi-dimensional probability space. Then, the mathematical model of the proposed Ps-rq strategy can be established to obtain the QoS metrics through the steady-state solution and the performance of the secondary system can be evaluated.

4.1. Scheduling Dynamic Analysis and CTMC Modeling

Note that a general state is as follows:

$$x = \{l_w, \dots, l_v, g_a, g_b, l_{pu}, l_{eq}, l_{rq}, l_{hq}\}$$

where l_m ($w \leq m \leq v$) is the number of SU_e with m aggregated channels, g_a is the number of SU_r with a aggregated channels, g_b is the number of SU_e^h with b aggregated channels, l_{pu} is the number of PUs in the system and l_{eq} , l_{rq} and l_{hq} are the current queue lengths of L_e , L_r and L_e^h , respectively.

Further, the set of feasible states of the system can be noted as

$$S = \{l_w, \dots, l_v, g_a, g_b, l_{pu}, l_{eq}, l_{rq}, l_{hq} \geq 0; b(x) \leq M; l_{eq} \leq Q_{eq}; l_{rq} \leq Q_{rq}; l_{hq} \leq Q_{hq}; \sum_{i=1}^{v-w} i l_{w+i} < w, \text{ if } l_{eq} > 0\} \quad (4)$$

Here, Q_{eq} , Q_{rq} and Q_{hq} are the total capacities of L_e , L_r and L_e^h , respectively. $\sum_{i=1}^{v-w} i l_{w+i} < w$, if $l_{eq} > 0$ means that if L_e is not empty, the summation of the maximum number of channels

that all ongoing SU_e can donate will always be less than its minimum aggregation number. $b(x)$ is the total number of the utilized channels at state x that can be described by:

$$b(x) = l_{pu} + ag_a + bg_b + \sum_{m=w}^v ml_m \quad (5)$$

Furthermore, $M - b(x)$ is the number of the idle channels in the system (that is, N_{free} assumed in Scheme 1).

In order to analyze the transition rate from the general state x to other states, we assume that the user arrival in the system obeys the Poisson distribution and the departure obeys the exponential distribution. Note that the arrival rate of PU is λ_p and departure rate is μ_p . The arrival rates of SU_r and SU_e are λ_{sr} and λ_{se} , and the departure rates are μ_{sr} and μ_{se} , respectively. Moreover, the scheduling with feedback loop of SU_e may cause the external arrival of the services entering into L_e and L_e^h to no longer obey the Poisson distribution. To simplify our analysis, we assume that the setting of the feedback loop has little effect on the services in L_e and L_e^h , and suppose that the exogenous arrival of SU services all obey the Poisson distribution.

Based on the above assumptions, by analyzing the channel access process, the CTMC model based on the Ps-rq strategy is established. All possible destination states, activities, transition rates and transition conditions of the system starting from the general state x are developed.

- Event A: PU arrival

Upon the PU arrival, the state transitions of the secondary system starting from the general state x to all possible triggering events are derived as shown in Figure 4, and the transition conditions for all possible states are listed in Figure 5. Except for state t_1 in the figure, there are no idle channels in the system that can be directly allocated to the arrived PU. Thus, there exists $M - b(x) = 0$ in state t_2 to state t_{35} .

The arrival of the PU can pre-empt the service of any SUs. For state t_4 to state t_{14} and state t_{16} to state t_{17} of service interruption in Figure 5, it indicates that the SUs in these states cannot perform spectrum adaptation, i.e., $\forall m (w < m \leq v)$, $l_m = 0$. Since L_r has no feedback loop, the service of the interrupted SU_r will be forced to terminate, such as in state t_{30} to state t_{32} .

In state t_3 and state t_{15} , there may be two cases when the SUs do not perform spectrum adaptation: there is no secondary user in the system who can perform adaptation, that is $\forall m (w < m \leq v)$, $l_m = 0$, or the SUs in the system cannot perform spectrum adaptation and $\forall m (h = m = v)$, $l_m > 0$. Similar conditions must be met for states t_{18} , t_{30} and t_{33} .

In addition, since $a = b < w$, $a = b \leq w - 1$. That is, the channels released by the interrupted SU_e cannot start the service of SU_e in L_e but the service of SU_r and SU_e^h in L_r and L_e^h can be started, such as in state t_6 to state t_{14} . This is the same as the channels released by the forcibly terminated SU_e , such as in state t_{21} to state t_{29} . However, the channels released by the interrupted or the terminated SU_r and SU_e^h cannot support the start of any service of SUs in the queue.

- Event B: PU departure

Upon the PU departure, the state transitions of the secondary system starting from the general state x to all possible triggering events are derived in Figure 6, and the transition conditions for all possible states are listed in Figure 7.

In this case, if the queue L_e^h is not empty and $b = 1$, then the service of SU_e^h in the queue can be started directly. If $b > 1$, then $M - b(x) = b - 1$ is required, that is, there exist $b - 1$ channels in the system, such as in state t_1 . This is the same as SU_r such as that in state t_2 . Since $a = b < w$, i.e., $w \neq 1$, the service of SU_e in L_e can only be started under the condition of $M - b(x) = w - 1$, such as in state t_3 .

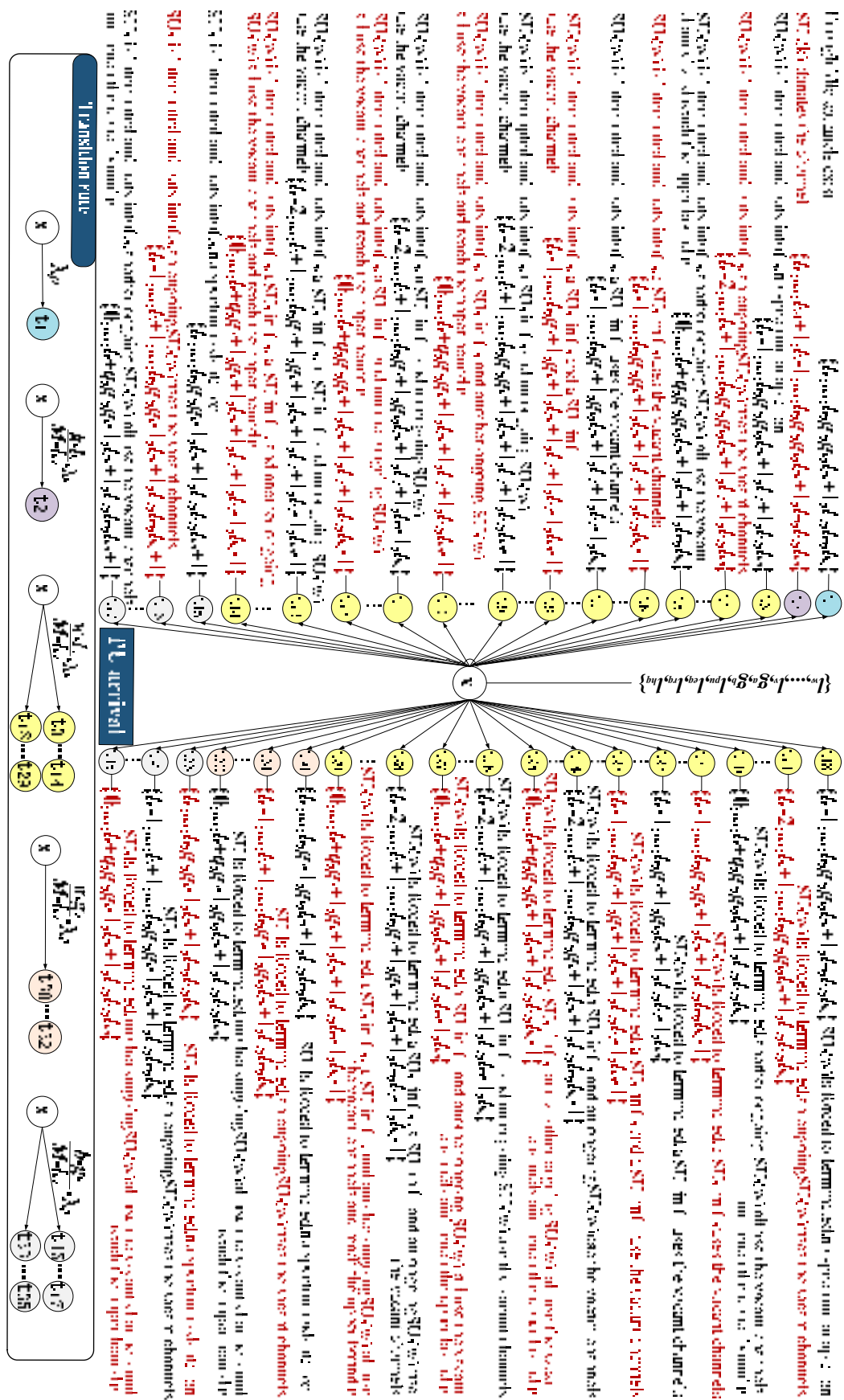


Figure 4. The transition of the system from the general state x to all reachable states on PU arrival.

State	Transition Conditions	State	Transition Conditions
t1	$M - b(x) > 0$	t18	$M - b(x) = 0; l_w > 0; l_{eq} = Q_{eq}; \left\{ \begin{array}{l} \forall m(w = m = v), l_m > 0 \\ \text{or} \\ \forall m(w < m \leq v), l_m = 0 \end{array} \right.$
t2	$M - b(x) = 0; l_k > 0; k = \max\{w+1, \dots, v\}$	t19	$M - b(x) = 0; l_w > 1; l_{eq} = Q_{eq}; m = 2w-1 \leq v; l_{rq} = l_{hq} = 0;$ $\forall m(w < m \leq v), l_m = 0;$
t3	$M - b(x) = 0; l_w > 0; l_{eq} < Q_{eq}; \left\{ \begin{array}{l} \forall m(w = m = v), l_m > 0 \\ \text{or} \\ \forall m(w < m \leq v), l_m = 0 \end{array} \right.$		
t4	$M - b(x) = 0; l_w > 1; l_{eq} < Q_{eq}; m = 2w-1 \leq v;$ $l_{rq} = l_{hq} = 0; \forall m(w < m \leq v), l_m = 0;$	t20	$M - b(x) = 0; l_w > 1; l_{eq} = Q_{eq}; q = l_w - 1; l_{rq} = l_{hq} = 0;$ $w \geq (v-w)(l_w-1)+1; \forall m(w < m \leq v), l_m = 0; 1 < w < v;$
.....		t21	$M - b(x) = 0; l_{eq} = Q_{eq}; l_{hq} > 0; w-1 \geq b;$ $\forall m(w < m \leq v), l_m = 0;$
t5	$M - b(x) = 0; l_w > 1; l_{eq} < Q_{eq}; q = l_w - 1; 1 < w < v;$ $w \geq (v-w)(l_w-1)+1; \forall m(w < m \leq v), l_m = 0; l_{rq} = l_{hq} = 0;$		
t6	$M - b(x) = 0; l_{eq} < Q_{eq}; l_{hq} > 0; w-1 \geq b;$ $\forall m(w < m \leq v), l_m = 0;$	t22	$M - b(x) = 0; l_{eq} = Q_{eq}; l_{rq} > 0; w-1 \geq a; l_{hq} = 0;$ $\forall m(w < m \leq v), l_m = 0;$
.....			
t7	$M - b(x) = 0; l_{eq} < Q_{eq}; l_{rq} > 0; w-1 \geq a; l_{hq} = 0;$ $\forall m(w < m \leq v), l_m = 0;$	t23	$M - b(x) = 0; l_w > 0; l_{eq} = Q_{eq}; l_{rq} > 0; l_{hq} > 0;$ $w-1 \geq a+b; \forall m(w < m \leq v), l_m = 0;$
.....			
t8	$M - b(x) = 0; l_w > 0; l_{eq} < Q_{eq}; l_{rq} > 0; l_{hq} > 0;$ $w-1 \geq a+b;$	t24	$M - b(x) = 0; l_{eq} = Q_{eq}; l_{hq} > 0; l_w > 1; 1 < w < v;$ $b < w-1 = k-w+b < b+a; \forall m(w < m \leq v), l_m = 0;$
.....			
t9	$M - b(x) = 0; l_{eq} < Q_{eq}; l_{hq} > 0; l_w > 1; 1 < w < v;$ $b < w-1 = k-w+b < b+a; \forall m(w < m \leq v), l_m = 0;$	t25	$M - b(x) = 0; l_{eq} = Q_{eq}; l_{hq} > 0; b < w-1 = b+(v-w)l_w-1;$ $q = l_w-1; l_w > 1; \forall m(w < m \leq v), l_m = 0; 1 < w < v;$
.....			
t10	$M - b(x) = 0; l_{eq} < Q_{eq}; l_{hq} > 0; b < w-1 = b+(v-w)l_w-1;$ $q = l_w-1; l_w > 1; \forall m(w < m \leq v), l_m = 0; 1 < w < v;$	t26	$M - b(x) = 0; l_{eq} = Q_{eq}; l_{rq} > 0; l_w > 1; b < w-1 = k-w+a;$ $l_{hq} = 0; \forall m(w < m \leq v), l_m = 0; 1 < w < v;$
.....			
t11	$M - b(x) = 0; l_{eq} < Q_{eq}; l_{rq} > 0; l_w > 1; b < w-1 = k-w+a;$ $l_{hq} = 0; 1 < w < v; \forall m(w < m \leq v), l_m = 0;$	t27	$M - b(x) = 0; l_{eq} = Q_{eq}; l_{rq} > 0; b < w-1 = a+(v-w)l_w-1; l_{hq} = 0;$ $l_w > 1; q = l_w-1; \forall m(w < m \leq v), l_m = 0; 1 < w < v;$
.....			
t12	$M - b(x) = 0; l_{eq} < Q_{eq}; l_{rq} > 0; b < w-1 = a+(v-w)l_w-1;$ $l_{hq} = 0; q = l_w-1; l_w > 1; \forall m(w < m \leq v), l_m = 0; 1 < w < v;$	t28	$M - b(x) = 0; l_{eq} = Q_{eq}; l_w > 1; l_{rq} > 0; l_{hq} > 0; w-1 = a+b+k-w;$ $\forall m(w < m \leq v), l_m = 0; 1 < w < v;$
.....			
t13	$M - b(x) = 0; l_{eq} < Q_{eq}; l_w > 1; l_{rq} > 0; l_{hq} > 0;$ $w-1 = a+b+k-w; 1 < w < v; \forall m(w < m \leq v), l_m = 0;$	t29	$M - b(x) = 0; l_{eq} = Q_{eq}; l_{hq} > 0; l_{rq} > 0; q = l_w-1; l_w > 1;$ $w-1 = a+b+(v-w)(l_w-1); \forall m(w < m \leq v), l_m = 0; 1 < w < v;$
.....			
t14	$M - b(x) = 0; l_{eq} < Q_{eq}; l_{hq} > 0; l_{rq} > 0; q = l_w-1; l_w > 1;$ $w-1 = a+b+(v-w)(l_w-1); 1 < w < v; \forall m(w < m \leq v), l_m = 0;$	t30	$M - b(x) = 0; g_a > 0; \left\{ \begin{array}{l} \forall m(w = m = v), l_m > 0 \\ \text{or} \\ \forall m(w < m \leq v), l_m = 0 \end{array} \right.$
.....			
t15	$M - b(x) = 0; g_b > 0; l_{hq} < Q_{hq}; \left\{ \begin{array}{l} \forall m(w = m = v), l_m > 0 \\ \text{or} \\ \forall m(w < m \leq v), l_m = 0 \end{array} \right.$	t31	$M - b(x) = 0; g_a > 0; k = a-1; 1 < w < k \leq v; l_w > 0;$ $\forall m(w < m \leq v), l_m = 0;$
.....			
t16	$M - b(x) = 0; g_b > 0; l_{hq} < Q_{hq}; k = b-1; 1 < w < k \leq v;$ $l_w > 0; \forall m(w < m \leq v), l_m = 0;$	t32	$M - b(x) = 0; g_a > 0; q = l_w; a-1 = (v-w)l_w; l_w > 0;$ $\forall m(w < m \leq v), l_m = 0; 1 < w < v;$
.....			
t17	$M - b(x) = 0; g_b > 0; l_{hq} < Q_{hq}; q = l_w; b-1 = (v-w)l_w;$ $1 < w < v; l_w > 0; \forall m(w < m \leq v), l_m = 0;$	t33	$M - b(x) = 0; g_b > 0; l_{hq} = Q_{hq}; \left\{ \begin{array}{l} \forall m(w = m = v), l_m > 0 \\ \text{or} \\ \forall m(w < m \leq v), l_m = 0 \end{array} \right.$
.....			
		t34	$M - b(x) = 0; g_b > 0; l_{hq} = Q_{hq}; k = b-1; 1 < w < k \leq v;$ $l_w > 0; \forall m(w < m \leq v), l_m = 0;$
			
		t35	$M - b(x) = 0; g_b > 0; l_{hq} = Q_{hq}; q = l_w; b-1 = (v-w)l_w;$ $l_w > 0; \forall m(w < m \leq v), l_m = 0; 1 < w < v;$

Figure 5. The transition conditions of the system from the general state x to all reachable states on PU arrival.

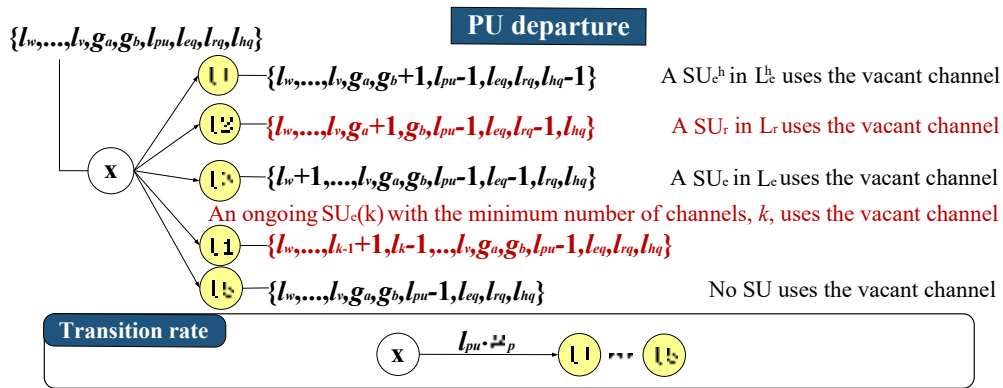


Figure 6. The transition of the system from the general state x to all reachable states on PU departure.

State	Transition Conditions	State	Transition Conditions
$t1$	$l_{pu} > 0; l_{hq} > 0; \begin{cases} b=1 \\ \text{or} \\ b > 1, M-b(x)=b-1 \end{cases}$	$t4$	$l_{pu} > 0; l_k > 0; k = \min\{w, \dots, v-1\}; 1 < w < v;$ $\begin{cases} l_{rq}=0 \\ \text{or} \\ l_{rq} > 0, M-b(x)+1 < a, a > 1 \end{cases}$ or $\begin{cases} l_{eq}=0 \\ \text{or} \\ l_{eq} > 0, M-b(x)+1 < w-1 \end{cases}$
$t2$	$l_{pu} > 0; l_{rq} > 0; l_{hq} = 0;$ $\begin{cases} a=1 \\ \text{or} \\ a > 1, M-b(x)=a-1 \end{cases}$	$t5$	$l_{pu} > 0;$ $\begin{cases} l_{hq}=0 \\ \text{or} \\ l_{hq} > 0, M-b(x)+1 < b, b > 1 \end{cases}$ or $\begin{cases} l_{rq}=0 \\ \text{or} \\ l_{rq} > 0, M-b(x)+1 < a, a > 1 \end{cases}$
$t3$	$l_{pu} > 0; l_{eq} > 0; l_{rq} = l_{hq} = 0;$ $M-b(x) = w-1;$		$\begin{cases} l_{eq}=0 \\ \text{or} \\ l_{eq} > 0, M-b(x)+1 < w-1 \end{cases}$ or $\begin{cases} \text{if } l_k > 0, w=k=v \\ \text{if } l_k=0, w \leq k \leq v (k \neq v) \end{cases}$

Figure 7. The transition conditions of the system from the general state x to all reachable states on PU departure.

- Event C: SU arrival

Upon the SU arrival, the state transitions of the secondary system starting from the general state x to all possible triggering events are derived as shown in Figure 8, and the transition conditions for all possible states are listed in Figure 9.

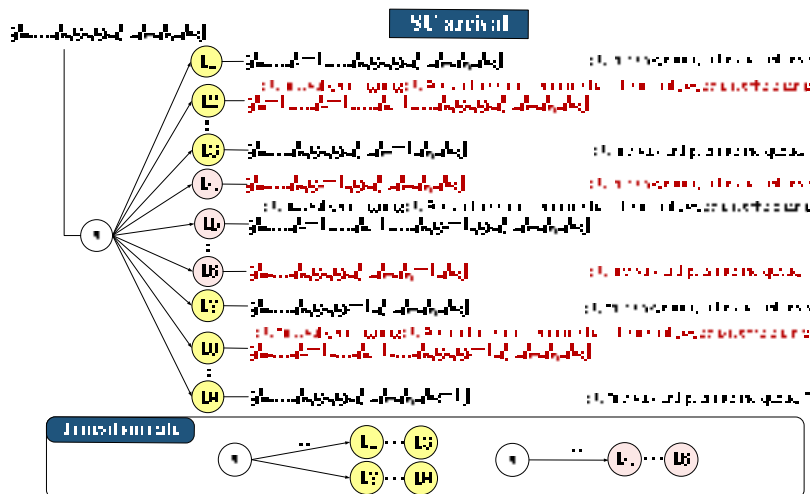


Figure 8. The transition of the system from the general state x to all reachable states on SU arrival.

State	Transition Conditions	State	Transition Conditions
t1	$M - b(x) > w;$	t6	$l_{rq} < Q_{rq}; \begin{cases} \text{if } w < v, M - b(x) + \sum_{m=w+1}^v (m-w)l_m < a; \\ \text{if } w = m = v, M - b(x) < a \end{cases}$
t2	$l_m > 0; m = \max\{w+1, \dots, v\}; w < v;$ $m - h + M - b(x) = w;$	t7	$M - b(x) > b;$
.....		t8	$l_m > 0; m = \max\{w+1, \dots, v\}; w < v;$ $m - h + M - b(x) = b;$
t3	$l_{eq} < Q_{eq}; \begin{cases} \text{if } w < v, M - b(x) + \sum_{m=w+1}^v (m-w)l_m < w; \\ \text{if } w = m = v, M - b(x) < w \end{cases}$	t9	$l_{hq} < Q_{hq}; \begin{cases} \text{if } w < v, M - b(x) + \sum_{m=w+1}^v (m-w)l_m < b; \\ \text{if } w = m = v, M - b(x) < b \end{cases}$
t4	$M - b(x) > a;$		
t5	$l_m > 0; m = \max\{w+1, \dots, v\}; w < v;$ $m - h + M - b(x) = a;$		

Figure 9. The transition conditions of the system from the general state x to all reachable states on SU arrival.

The condition of the service of SU_e can be started when its arrival is the sum of the number of the idle channels existing in the system and the channels that can be donated by the ongoing SUs are not less than the lower bound of SU_e , i.e., $M - b(x) + \sum_{m=w+1}^v (m-w)l_m \geq w$. Otherwise, it will be put into the queue L_e , provided that $l_{eq} < Q_{eq}$, such as in state t_3 . SU_r and SU_e^h arrive in the same way, i.e., state t_6 and state t_9 .

- Event D: SU departure

Upon the SU departure, the state transitions of the secondary system starting from the general state x to all possible triggering events are derived as shown in Figure 10, and the transition conditions for all possible states are listed in Figure 11.

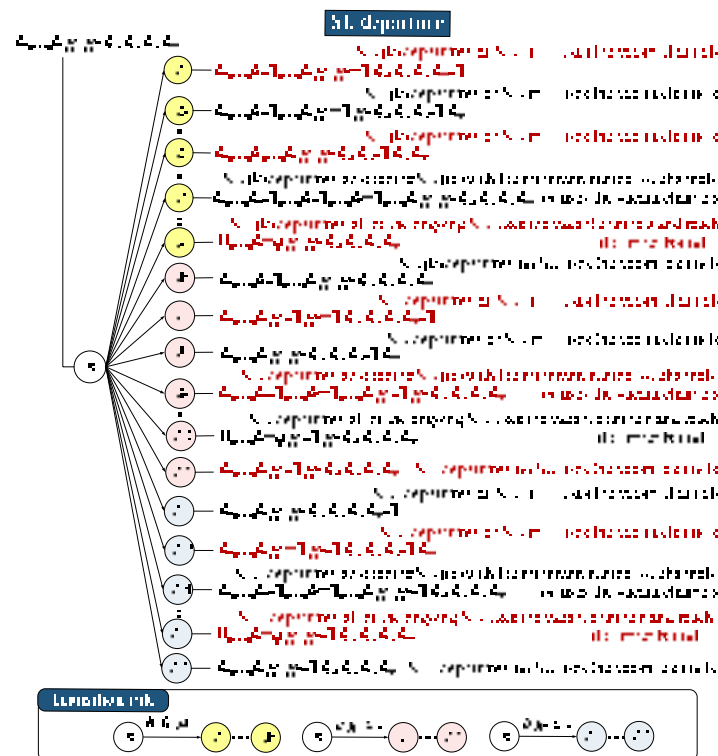


Figure 10. The transition of the system from the general state x to all reachable states on SU departure.

State	Transition Conditions	State	Transition Conditions
t1	$l_k > 0; l_{hq} > 0; k \geq b;$	t9	$g_a > 0; l_m > 0; w < h = m + a \leq v;$ $l_{rq} = l_{hq} = 0;$
t2	$l_k > 0; l_{rq} > 0; l_{hq} = 0; k \geq a;$		
.....		t10	$g_a > 0; q = \sum_{k=w}^{v-1} l_k; a \geq \sum_{k=w}^{v-1} (v-k)l_k;$ $l_{rq} = l_{hq} = 0; w < v;$
t3	$l_k > 0; l_{eq} > 0; l_{rq} = l_{hq} = 0;$		
.....		t11	$g_a > 0; l_{rq} = l_{hq} = 0; \begin{cases} l_m = 0, w < m \leq v; \\ or \\ l_m > 0, w = m = v; \end{cases}$
t4	$l_m > 0; l_k > 0; w < h = m + k \leq v;$ $m = \max\{w, \dots, v\}; l_{rq} = l_{hq} = l_{eq} = 0;$	t12	$g_b > 0; l_{hq} > 0;$
.....		t13	$g_b > 0; l_{rq} > 0; l_{hq} = 0;$
t5	$l_k > 0; q = \sum_{m=w}^{v-1} l_m - 1; l_{rq} = l_{hq} = l_{eq} = 0;$ $k \geq \sum_{m=w}^{v-1} (v-m)l_m - (v-k); w < v;$	t14	$g_b > 0; l_m > 0; w < h = m + b \leq v; l_{rq} = l_{hq} = 0;$
t6	$l_k > 0; l_{rq} = l_{hq} = l_{eq} = 0;$ $\begin{cases} l_m = 0, w < m \leq v; \\ or \\ l_m > 0, w = m = v; \end{cases}$		
t7	$g_a > 0; l_{hq} > 0;$	t15	$g_b > 0; q = \sum_{k=w}^{v-1} l_k; b \geq \sum_{k=w}^{v-1} (v-k)l_k;$ $w < v; l_{rq} = l_{hq} = 0;$
t8	$g_a > 0; l_{rq} > 0; l_{hq} = 0;$	t16	$g_b > 0; l_{rq} = l_{hq} = 0; \begin{cases} l_m = 0, w < m \leq v; \\ or \\ l_m > 0, w = m = v; \end{cases}$

Figure 11. The transition conditions of the system from the general state x to all reachable states on SU departure.

In this case, the total number of idle channels used by the ongoing SUs should not be greater than the number of channels released by the departing SUs, such as the condition of $k \geq \sum_{m=w}^{v-1} (v-m)l_m - (v-k)$ in state t_5 . This is similar to that in state t_{10} and state t_{15} .

4.2. QoS Metrics

Through CTMC modeling, any possible user activity may trigger the transition from any reachable state t_i to another reachable state t_j , as shown in Figures 4–11. q_{ij} ($i, j \in S$ and $i \neq j$) represents the total transition rate, where S is the system's feasible state set. Let $\pi(x)$ be the steady-state probability of the system, which can be calculated from global balance equations and the normalization condition as follows.

$$\pi(x)Q = 0, \sum_x \pi(x) = 1 \quad (6)$$

where $\pi(x)$ is the steady-state probability vector and Q is the transition rate matrix generated by the transition rate q_{ij} . The diagonal elements of transition rate matrix Q , q_{ii} can be found as $q_{ii} = -\sum_{j \neq i} q_{ij}$. Thus, the QoS metrics of the secondary system based on the proposed Ps-rq strategy can be obtained.

- Network capacity refers to the average number of SU services completed in CRNs per time unit (unit: packet/unit time) [12]. Let C_e , C_r be the capacities of SU_e , SU_r , respectively, we can get:

$$C_r = \sum_{x \in S} a g_a \mu_{sr} \pi(x) \quad (7)$$

$$C_e = \sum_{x \in S} \sum_{k=w}^v (k l_k + b g_b) \mu_{se} \pi(x) \quad (8)$$

where $x \in S$ is the set of all possible states in the systems contained in Figures 4–11.

- Spectrum utilization refers to the ratio of the average number of utilized channels to the total number of available channels [25], which can be determined by:

$$U = \sum_{x \in S} \frac{b(x)}{M} \pi(x) \quad (9)$$

where $b(x)$ is the total number of available channels of state x , M is the total number of channels in the system and S is the set of all possible transition states contained in Figures 4–11.

- Blocking probability refers to the probability that the newly arrived SU cannot be served due to all of the channels in the system being busy. Since the waiting queue is set for SUs in this study, the service of the newly arrived SU will be blocked when the corresponding queue is full. It should be noted that the SU_e^h entering L_e^h is the SU_e interrupted by the PU, but not the SUs who are newly arriving in the system. Therefore, the SU_e^h does not have the case of being blocked.

The newly arrived SU_e will be blocked if both of the following conditions are met: (1) the sum of the total number of idle channels in the system and the total number of channels that can be donated by the ongoing SU_e are not enough to start the service of the SU_e (that is the lower bound of channel assembling is not reached) and (2) the queue, L_e , is full. Thus, the blocking probability of SU_e , P_b^e , can be obtained by:

$$P_b^e = \sum_{A_e} \pi(x) \quad (10)$$

where the set A_e is the trigger condition for which SU_e is blocked:

$$A_e = \left\{ x \in S, l_{eq} = Q_{eq}, M - b(x) + \sum_{k=w+1}^v (k-w)l_k < w \right\}$$

Similarly, the newly arrived SU_r will be blocked if both of the following conditions are met: (1) the sum of the total number of idle channels in the system and the total number of channels that can be donated by the ongoing SU_e are not enough to start the service of the SU_r and (2) the queue, L_r , is full. Then, the blocking probability of SU_r , P_b^r , can be determined by:

$$P_b^r = \sum_{A_r} \pi(x) \quad (11)$$

where the set A_e is the trigger condition for which SU_e is blocked:

$$A_r = \left\{ x \in S, l_{rq} = Q_{rq}, M - b(x) + \sum_{k=w+1}^v (k-w)l_k < a \right\}$$

- Forced termination probability refers to the probability that the SUs being served are forced to terminate due to the arrival of a PU. As the set of the feedback loop of SU_e , the SU_e interrupted by the PU can return to the queue L_e and L_e^h through packet classifier II. If the corresponding queue is full, the service will be forced to terminate.

Accordingly, the service of SU_e^h is forced to terminate when the following three conditions are met: (1) there are no idle channels in the system; (2) the ongoing SU_e cannot perform spectrum adaptation to share channels with the interrupted SUs and (3) the queue, L_e^h , is full. To be specific, states t_{33} to t_{35} in Figure 4 indicate that the PU pre-empts any SU_e^h with b aggregated channels, and the SU_e^h is forced to terminate. The forced termination

probability of SU_e^h , P_f^{eh} , can be calculated by the ratio of the average forced termination probability to the arrival rate of SU_e^h , λ_{se} , as follows:

$$P_f^{eh} = \sum_{B_e^h} \frac{\lambda_p b g_b}{(M - l_{pu}) \lambda_{se}} \pi(x) \quad (12)$$

where B_e^h is the set of conditions for which SU_e^h is forced to terminate:

$$B_e^h = \left\{ x \in S, M = b(x), l_{pu} < M, g_b > 0, l_{hq} = Q_{hq}, \left\{ \begin{array}{l} \forall m(w < m \leq v), l_m = 0 \\ \text{or} \\ \forall m(w = m = v), l_m > 0 \end{array} \right\} \right\}$$

Similarly, the service of SU_e is forced to terminate when the following three conditions are met: (1) after being pre-empted, the number of remaining channels of SU_e is insufficient to start the service, that is the SU_e with w aggregated channels is pre-empted; (2) the ongoing SU_e cannot perform spectrum adaptation to share channels with the interrupted SUs and (3) the queue, L_e , is full. On this basis, states t_{18} to t_{29} in Figure 4 indicate that the PU pre-empts any SU_e with w assembled channels, and the SU_e is forced to terminate. The forced termination probability of SU_e , P_f^e , can be obtained by:

$$P_f^e = \sum_{B_e} \frac{\lambda_p w l_w}{(M - l_{pu}) \Lambda_e} \pi(x) \quad (13)$$

where Λ_e represents the arrival rate of SU_e that is not blocked: $\Lambda_e = \lambda_{se}(1 - P_b^e)$ and B_e is the set of conditions for which SU_e is forced to terminate:

$$B_e = \left\{ x \in S, M = b(x), l_{pu} < M, l_w > 0, l_{eq} = Q_{eq}, \left\{ \begin{array}{l} \forall m(w < m \leq v), l_m = 0 \\ \text{or} \\ \forall m(w = m = v), l_m > 0 \end{array} \right\} \right\}$$

Since the feedback loop is not set for the queue L_r , if the following two conditions are met, the service of SU_r is forced to terminate: (1) there are no idle channels in the system to continue its service and (2) the ongoing SU_e cannot donate the channels for it. Based on this, states t_{30} to t_{32} in Figure 4 indicate that the PU pre-empts any SU_r with a aggregated channels, and the SU_r is forced to terminate. The forced termination probability of SU_r , P_f^r , can be obtained by:

$$P_f^r = \sum_{B_r} \frac{\lambda_p a g_a}{(M - l_{pu}) \Lambda_r} \pi(x) \quad (14)$$

where Λ_r represents the arrival rate of SU_r that is not blocked: $\Lambda_r = \lambda_{sr}(1 - P_b^r)$ and B_r is the set of conditions for which SU_r is forced to terminate:

$$B_r = \left\{ x \in S, M = b(x), l_{pu} < M, g_a > 0, \left\{ \begin{array}{l} \forall m(w < m \leq v), l_m = 0 \\ \text{or} \\ \forall m(w = m = v), l_m > 0 \end{array} \right\} \right\}$$

5. Performance Analysis of the Proposed Ps-rq Strategy

5.1. Model Characteristics Comparison and Analysis

In this section, we compare the proposed Ps-rq strategy with two classical channel allocation schemes (reference [12], 2014) as well as two novel ones (reference [18], 2020 and reference [9], 2022), aiming to illustrate the characteristics of the proposed strategy. The comparison of model characteristics is shown in Figure 12.

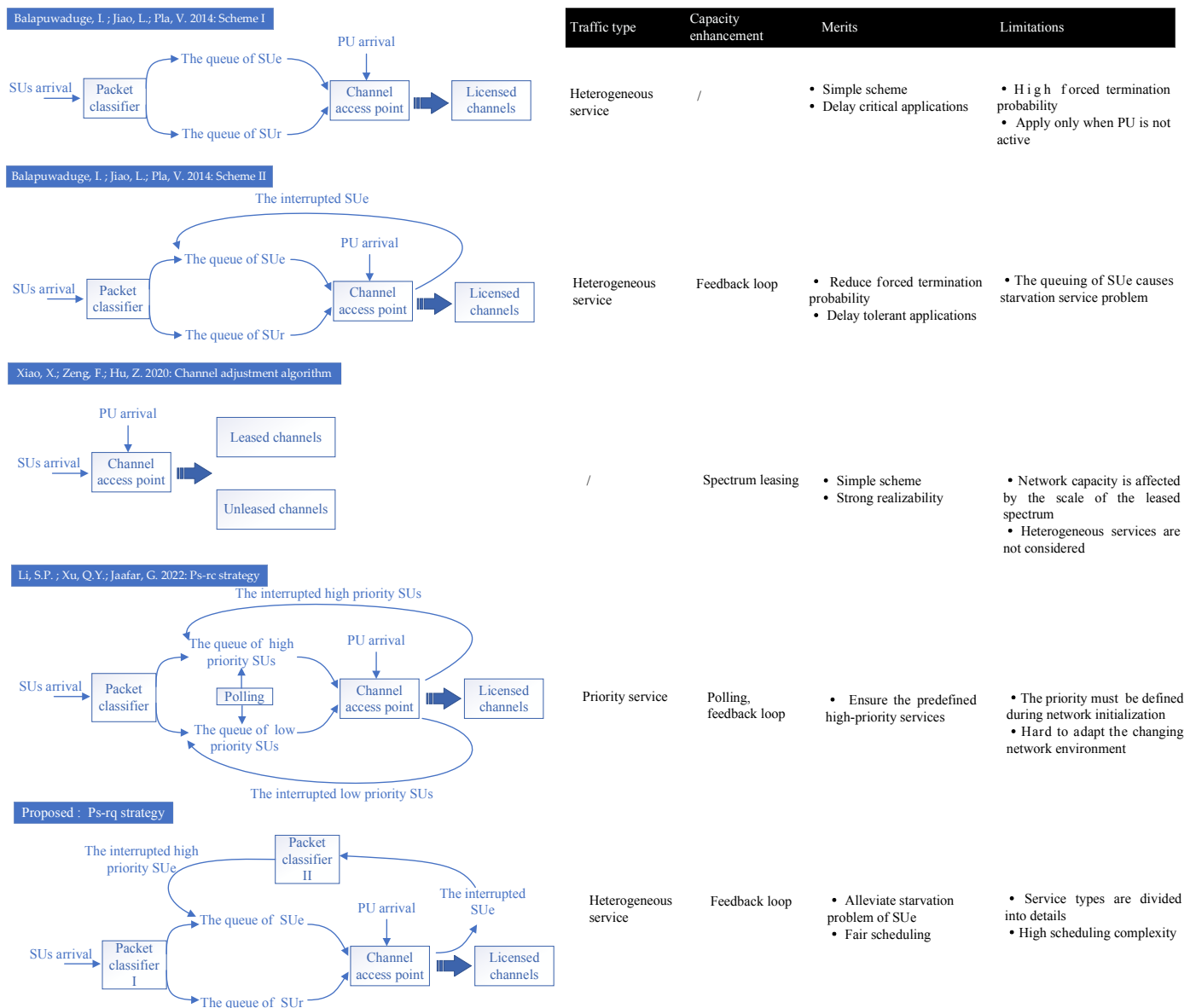


Figure 12. The comparison of model characteristics [9,12,18].

The mentioned research models, as depicted in Figure 12, aim to enhance the capacity of the secondary networks. By employing the random analysis method and establishing the corresponding continuous-time Markov chain for systematic analysis, performance metrics are derived to evaluate the effectiveness of the research schemes. However, the mentioned studies have faced challenges in distinguishing service types, as demonstrated in [9,18], or encountered limitations in implementing differentiated service applications due to network environment constraints, such as that in [12].

These limitations serve as the inspiration for the strategy outlined in this paper. The Ps-rq strategy proposed in this paper aims to provide a more equitable differentiation service by addressing the issue of starvation. To achieve this, we employ packet classifiers I and II to perform two levels of service differentiation. The purpose of packet classifier I is to divide the arriving SUs into SU_e and SU_r . Considering that if the SU_e interrupted by the PU is all queued for re-service, it will inevitably lead to the starvation problem of another queue. The model proposed in this paper, therefore, considers the service requirements of the partially interrupted high-priority SU_e and incorporates packet classifier 2 to ensure fair service provision while offering differentiated services, which represents a notable aspect

of our research framework. However, the complexity of the proposed scheduling strategy remains challenging to circumvent in light of service classification refinement. Next, we will conduct a detailed performance analysis of the research Ps-rq strategy presented in this paper.

5.2. Numerical Simulation and Analysis

To illustrate the performance of the secondary system based on the proposed channel assembling strategy, in this section, numerical simulation results of elastic services and real-time services are presented based on MATLAB. The simulation results are based on the procedure for the dimension reduction of high-dimensional Markov chains. In order to verify the developed model, we present the simulation results of the secondary system performance under varying primary user arrival rates. The basic parameters of the numerical simulation are set as follows.

The number of the PUs in the system is set to 6 ($M = 6$), and the numerical experiment is carried out in a CRN with six channels sharing the spectrum bandwidth. The arrival rate of the PU is set as $\lambda_p = 1$, and the departure rate is set as $\mu_p = 0.5$. The arrival rate and departure rate of the elastic service are set as $\lambda_{se} = 2$ and $\mu_{se} = 1$, respectively, and that of the real-time service are set as $\lambda_{sr} = 1$ and $\mu_{sr} = 1$. Note that the unit of these parameters is “sessions per time unit”. By assigning specific values to these parameters, such as 0.5 sessions/sec and flow length in bits, the capacity can be quantified in terms of kbps or Mbps.

The dynamic parameters of the system $(a, b, w, v, Q_{eq}, Q_{rq}, Q_{hq})$ in the spectrum access process are set as $a = b = 1$, $w = 2$, $v = 4$, $Q_{eq} = 2$, $Q_{rq} = 1$ and $Q_{hq} = 1$. The proportional partition coefficient is $\alpha : \beta = 1 : 1$. Based on the above system settings, the general state x of the system in Section 4.1 can be specified:

$$(l_2^3 l_3^2 l_4^1 g_1^6 g_2^6 l_{pu}^2 Q_{eq}^1 Q_{rq}^1 Q_{hq}^1) \quad (15)$$

where l_i ($i = 2, 3, 4$) represents SU_e who aggregates i channels and g_a, g_b represent SU_r with a aggregated channels or SU_e^h with b aggregated channels ($a = b = 1$); the superscript of each position indicates the upper limit of the number of users at the corresponding position, such as l_3^2 , which indicates that there are at most two SU_e assembling three channels in the system at the same time, l_{pu}^6 , which means that at most six PUs exist in the system at the same time and Q_{eq}^2 , which indicates that the maximum capacity of the queue L_e is 2.

Numerical simulation is based on the parameterized general state. The procedure for the dimension reduction of high-dimensional Markov chains is presented as follows. Since the general state assumed in this paper is high-dimensional, it is difficult to directly generate the transition rate matrix Q . Therefore, we consider downgrading the markers of high-dimensionality. Theoretically, the size of the matrix Q is 2592×2592 , which is generated by the product of the superscripts of Equation (15). Take any general state $I : (1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0)$ existing in the system to show the downgrade process as follows. The state $I : (1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0)$ indicates that the system has an ongoing SU_e with two aggregated channels, an ongoing SU_r with one aggregated channel, an ongoing SU_e^h with one aggregated channel and an ongoing PU. Therefore, there are five flows existing in the system in this state. Accordingly, state $I : (1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0)$ can be marked as state $I_5, -$, where the first position of the subscript represents the total number of co-existing flows in the system, and the second position is the counting in order, and the upper limit is the sum of the possible combination states with five flows in the system. If a user arrival occurs in the system in this state, such as a PU arrival, then the corresponding state transition is $(1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0) \rightarrow (1\ 0\ 0\ 1\ 1\ 2\ 0\ 0\ 0)$. In this case, there are six flows co-existing in the system and the transition process is marked as $I_5, - \rightarrow I_6, -$. If a user departure occurs in

the system in this state, such as an SU_e^h departure, then the corresponding state transition is $(1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0) \rightarrow (0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 0)$. In this case, there are four flows co-existing in the system and the transition process is marked as $I_{5,-} \rightarrow I_{4,-}$. According to Figures 4–11, the possible transition events and transition rates of the system in any general state can be obtained, and then the transition rate matrix Q can be developed. Furthermore, the steady-state probability $\pi(x)$ can be solved and the performance metrics of the secondary system including its network capacity, spectrum utilization, blocking probability and forced termination probability can be further simulated. We use MATLAB to solve the Markov model and to run simulations. On this basis, four study cases are presented and the channel assembling performance based on the proposed Ps-rq strategy is analyzed.

Case 1: capacity of secondary network vs. arrival rate of PUs

Figure 13 shows the relationship between the network capacity of SU_e and SU_r and the arrival rate of PUs. As can be seen from the figure, the capacity of the secondary network decreases with the increase in the arrival rate of PUs. When the number of PUs entering the system increases, the absolute data access priority will reduce the total number of available remaining channels in the system, so the number of SUs who can accept the services in the system will also decrease accordingly. As a result, the network capacity of SU_e and SU_r decreases with the increase in the arrival rate of PUs.

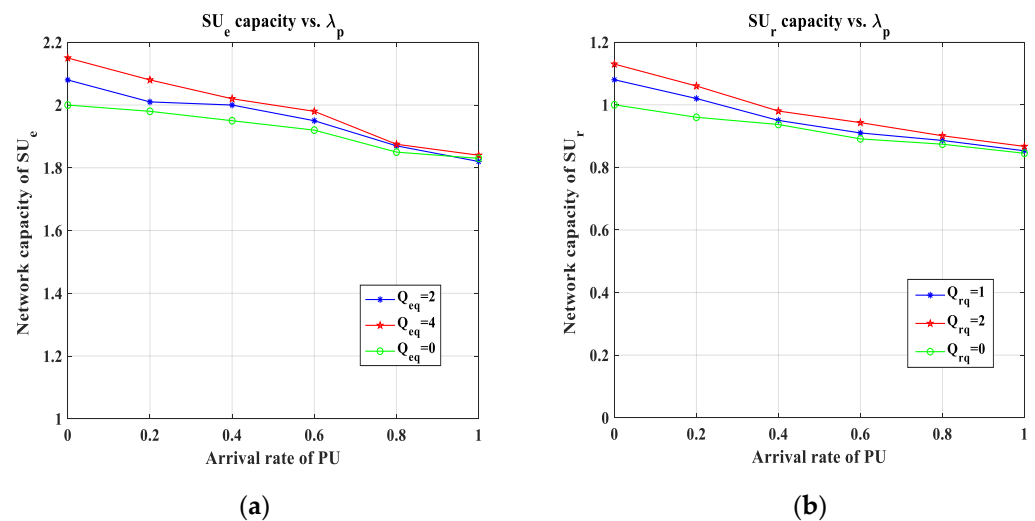


Figure 13. (a) Network capacity of SU_e ; (b) Network capacity of SU_r .

However, it can be seen from Figure 13 that increasing the total capacity of the queue of SUs increases the capacity of the secondary network. At $\lambda_p = 0.2$, the capacity of the queue of SU_e , Q_{eq} , increases from 0 to 4, and the secondary network capacity increases by 4.8%; the capacity of the queue of SU_r , Q_{rq} , increases from 0 to 4, which is a 9.43% increase in secondary network capacity. After the queue capacity is increased, the number of SU flows that can coexist in the system will increase, thus increasing the probability that SUs will be able to accept the service. By comparing Figure 13, it is not difficult to find that the gradient descent of the network capacity of SU_r is less than that of SU_e . This is because SU_r has a higher service priority than SU_e (except for the interrupted SU_e^h), so it is easier to start the service for the newly arrived SU_r than SU_e .

Case 2: spectrum utilization vs. arrival rate of SU_e

The relationship between spectrum utilization and the arrival rate of SUs is shown in Figure 14 with the arrival rate of SU_e as the x-axis. It can be seen from the figure that the spectrum utilization increases with the increase in the arrival rate. When $Q_{rq} = 2$, $Q_{eq} = 4$, the spectrum utilization rate is up to 92.3%. With the increase in the arrival rate of SUs, the idle spectrum is fully utilized. When the spectrum utilization becomes saturated

gradually, the gradient will decrease. In addition, it is not difficult to find that increasing the queue capacity can increase the secondary network capacity, so spectrum utilization is correspondingly improved.

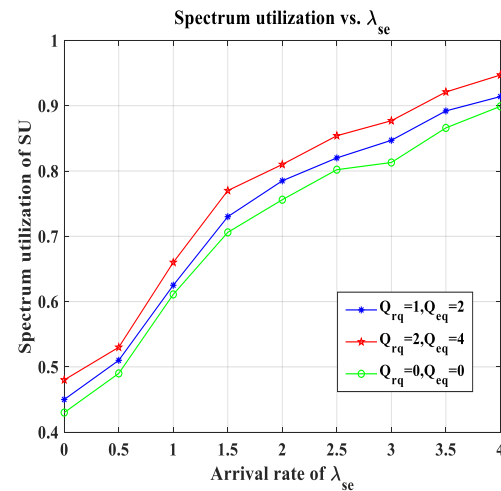


Figure 14. Spectrum utilization of SUs.

Case 3: blocking probability vs. arrival rate of PUs

Figure 15 shows the relationship between the blocking probability of SUs and the arrival rate of PUs. In general, the blocking probability of SUs increases with the arrival rate of PUs. The idle channels are fully utilized when the number of the PUs in the system increases due to the absolute data access priority, resulting in the increase in the blocking probability of SUs. Since SU_r has a higher priority than SU_e , the blocking probability of data access is lower than that of SU_e .

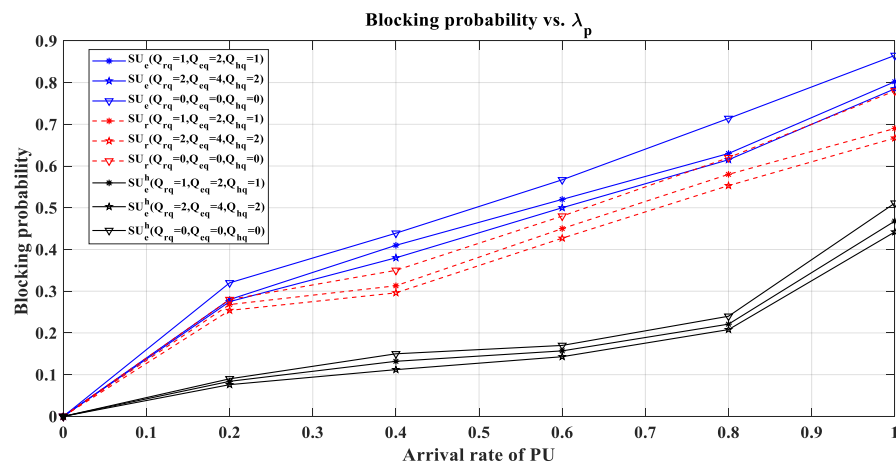


Figure 15. Blocking probability of SUs.

It appears that both curves are relatively stable, and the difference in blocking probability is no more than 0.1. It is illustrative of the fact that the proposed Ps-rq strategy can make SU_e and SU_r have a stable performance on the basis of reducing the blocking probability of high-priority users SU_e^h . The adoption of differentiated services provides a more equitable compromise scheduling without causing over-starvation of low-priority users (SU_e). However, the available idle spectrum existing in the secondary system decreases with the increase in the arrival rate of PUs, resulting in the blocking probability of SU_r and SU_e gradually approaching 1.

By comparing the blocking probability of SU_e and SU_r under different queue capacities, we can find that a larger capacity improves the reduction of blocking probability.

For example, for SU_r , when the arrival rate of PU is $\lambda_p = 0.6$, the blocking probability is reduced by 10.82% when the queue capacity is $Q_{rq} = 2$, $Q_{eq} = 4$ compared with $Q_{rq} = 0$, $Q_{eq} = 0$. And for SU_e , when $\lambda_p = 0.6$, the blocking probability is reduced by 11.04% under the same value. This explains that the system can provide services for more flows with the increase in the queue capacity, which provides more possibilities for the speculative access of SUs. By comparison, it can be observed that the increase in queue capacity can significantly improve the performance of SU_e . The reason is that a feedback loop is set for the service of SU_e in the proposed Ps-rq strategy and a larger queue capacity is installed at the system setup.

Case 4: forced termination probability vs. arrival rate of PUs

Figure 16 shows the relationship between the forced termination probability of SUs and the arrival rate of PUs. As is depicted in the figure, the forced termination probability of SUs increases with the arrival rate of PUs, which is also determined by the data access priority of PUs. Since a reserved queue is set for the interrupted SU_e^h , the forced termination probability of SU_e^h is the lowest in Figure 16, such as at the maximum difference value, $\lambda_p = 0.8$, when $Q_{rq} = 1$, $Q_{eq} = 2$, $Q_{hq} = 1$, the forced termination probability of SU_e^h is 29.23% lower than that of SU_e and 42.5% lower than that of SU_r . However, due to the setting of the feedback loop for SU_e , SU_r has a higher forced termination probability than SU_e . Therefore, the interrupted SU_e can go back to the queue and will not be forced to terminated until the queue is full.

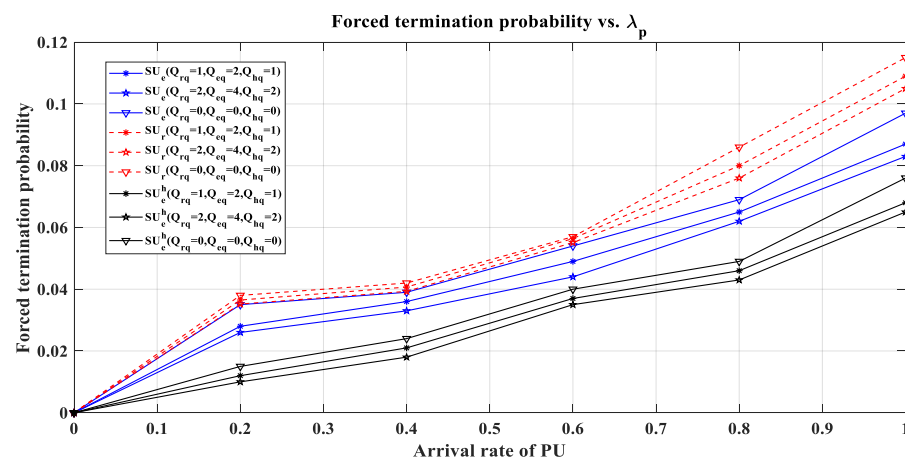


Figure 16. Forced termination probability of SUs.

Real-time services are more sensitive to delay, and packet loss caused by slight jitter does not cause serious interference to real-time voice/image quality. Therefore, no feedback loop is adopted for SU_r . The total channel occupancy rate can be reduced in this way, but at the expense of increasing the forced termination probability of SU_r . Taken overall, the proposed Ps-rq strategy can differentiate the service priorities of heterogeneous services and provide fair scheduling for them. On this basis, the service quality of the high-priority elastic service, SU_e^h , that is interrupted by PUs can be improved.

To further demonstrate the superiority of the proposed strategy in enhancing the performance of SUs, a comparison is provided between the Ps-rq strategy and the non-Ps-rq strategy regarding forced termination probability, as depicted in Figure 17. The comparison results all occurred under the initial conditions $Q_{eq} = 2$ and $Q_{rq} = 1$. As can be seen from the figure, whether for SU_e or SU_r , when the strategy is not adopted, the probability of forced termination for SUs increases significantly. The performance analysis of SU_e^h is specific to the strategy proposed in this paper; thus, there is no comparison of the forced termination probability of SU_e^h when not adopting Ps-rq. The figure illustrates that the increase in the forced termination probability for SU_e is greater than that for SU_r in the absence of Ps-rq. Especially when the arrival rate of PU is small, SU_e has more

opportunities to access the channel. It is evident from the proposed strategy that the excellent result can be attributed to the improved performance of SU_e^h .

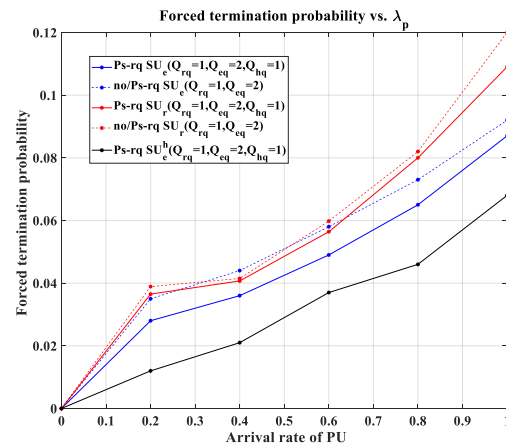


Figure 17. Comparison of forced termination probability with and without Ps-rq.

6. Conclusions

This study designs and proposes a priority scheduling strategy with reserved queue (Ps-rq strategy) for CRNs. This strategy adopts the service classification scheme combining static factors (message type) and dynamic factors (information validity, message correlation and message size), which can differentiate the elastic services with high-priority based on the heterogeneous service classification in secondary services. By setting a new reserved queue for vital elastic services, the proposed strategy can improve the service priority of high-priority elastic services interrupted by PUs. In addition, Takagi–Sugeno fuzzy control is used to prioritize elastic services, and a 3D priority view of elastic services is presented. On this basis, a proportional channel allocation scheme with priority that is combined with channel assembling technology and spectrum adaptation technology is provided based on the Ps-rq strategy. By dividing user activities in CRNs into four events including PU arrival, PU departure, SU arrival and SU departure, the flow of resources in the secondary network is investigated, and the dynamic channel access process based on the Ps-rq strategy is given. By establishing the CTMC, the resource flow process in the secondary network is mapped to the high-dimensional state space, and all possible transition states, transition rates and transition conditions triggered by the four kinds of events that start from a general state are derived. After downgrading the high-dimensional markers, the steady-state solution of the system can be obtained by acquiring the transition rate matrix, and a series of performance evaluation indexes such as secondary network capacity, spectrum utilization, blocking probability and forced termination probability are further obtained.

Furthermore, we compare the proposed Ps-rq strategy with four channel assembling schemes to demonstrate its suitability for specific scenarios involving the transmission of elastic traffic. By combining the procedure for dimension reduction of high-dimensional Markov chains, we numerically simulate the analytical results of performance indexes using MATLAB and provide four study cases. The simulation results indicate that, in the same dimension, as the buffer capacity increases, the SU_e 's network capacity increases by 4.8% and the blocking probability decreases by 11.04%, while the SU_r 's network capacity increases by 9.43% and the blocking probability decreases by 10.82%. And the forced termination probability of high-priority elastic services concerned in this paper is 29.23% lower than that of SU_e and 42.5% lower than that of SU_r . Numerical results show that the proposed strategy can help improve the service priority of high-priority elastic services that are interrupted by PUs in CRNs and reduce the forced termination probability on the basis of providing fair scheduling.

Author Contributions: Q.X. researched the literature, conceived the study concepts, provided the mathematical models, designed the algorithm, completed numerical simulations, took charge of the original draft preparation and edited the manuscript; S.L. conceived the study concepts, improved the systematic research and analysis methodology and supervised the completion of the refinement of the paper; J.G. and Y.H. gave valuable suggestions for revision and checked formula deducing and English grammar. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China (61663024) and the Xi'an Aeronautical University Natural Science doctoral fund project (2022KY0201).

Data Availability Statement: The data supporting this article are from previously reported studies and datasets, which have been cited.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Federal Communications Commission. *Spectrum Policy Task Force Report*. ET Docket; No. 02-135; Federal Communications Commission: Washington, DC, USA, 2002.
2. Huang, X.L.; Li, Y.X.; Gao, Y. Q-Learning-Based Spectrum Access for Multimedia Transmission Over Cognitive Radio Networks. *IEEE Trans. Cogn. Commun. Netw.* **2021**, *7*, 110–119. [[CrossRef](#)]
3. Srikar, D.; Nella, A.; Mamidi, R. A Novel Integrated UWB Sensing and 8-Element MIMO Communication Cognitive Radio Antenna System. *Electronics* **2023**, *12*, 330. [[CrossRef](#)]
4. Muzaffar, M.U.; Sharqi, R. A Review of Spectrum Sensing in Modern Cognitive Radio Networks. *Telecommun. Syst. Model. Anal. Des. Manag.* **2024**, *2*, 85. [[CrossRef](#)]
5. Ahmad, W.; Radzi, N.; Samidi, F.S. 5G Technology: Towards Dynamic Spectrum Sharing Using Cognitive Radio Networks. *IEEE Access* **2020**, *8*, 14460–14488. [[CrossRef](#)]
6. Azaly, N.; Badran, E.F. Performance Enhancement of Dynamic Spectrum Access via Channel Reservation for Cognitive Radio Networks. *Wirel. Pers. Commun.* **2021**, *118*, 2867–2883. [[CrossRef](#)]
7. Lei, J. *Channel Aggregation and Fragmentation for Traffic Flows*; Springer: Berlin/Heidelberg, Germany, 2020.
8. Liu, X.; Jia, M. Intelligent Spectrum Resource Allocation Based on Joint Optimization in Heterogeneous Cognitive Radio. *IEEE Trans. Emerg. Top. Comput. Intell.* **2020**, *4*, 5–12. [[CrossRef](#)]
9. Li, S.P.; Xu, Q.Y.; Jaafar, G. Modeling and Performance Analysis of Channel Assembling Based on Ps-rc Strategy with Priority Queues in CRNs. *Wirel. Commun. Mob. Comput.* **2022**, *17*, 6384261. [[CrossRef](#)]
10. Ma, L.; Xu, Y.; Fu, Y. A Resource Scheduling Scheme for Spectrum Aggregation in Cognitive Radio Based Heterogeneous Networks. *China Commun.* **2015**, *12*, 100–111.
11. Jiao, L.; Li, F.Y.; Pla, V. Modeling and Performance Analysis of Channel Assembling in Multichannel Cognitive Radio Networks with Spectrum Adaptation. *IEEE Trans. Veh. Technol.* **2012**, *61*, 2686–2697. [[CrossRef](#)]
12. Balapuwaduge, I.; Jiao, L.; Pla, V. Channel Assembling with Priority-Based Queues in Cognitive Radio Networks: Strategies and Performance Evaluation. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 630–645. [[CrossRef](#)]
13. Esenogho, E.; Srivastava, V.M. Two Heterogeneous Channel Assembling Strategies in Cognitive Radio Networks: A Performance Analysis. *Int. J. Eng. Technol. Innov.* **2017**, *7*, 98–116.
14. Zhang, W.; Sun, Y.; Deng, L. Dynamic Spectrum Allocation for Heterogeneous Cognitive Radio Networks with Multiple Channels. *IEEE Syst. J.* **2019**, *13*, 53–64. [[CrossRef](#)]
15. Zhu, X. Analysis of Cognitive Radio Spectrum Access with Optimal Channel Reservation. *IEEE Commun. Lett.* **2007**, *11*, 304–306. [[CrossRef](#)]
16. Esenogho, E.; Mambou, E.N. Integrating Queuing Regime into Cognitive Radio Channel Aggregation Policies: A Performance Evaluation. *J. Commun.* **2017**, 1–5. [[CrossRef](#)]
17. Esenogho, E.; Srivastava, V.M. Channel Assembling Strategy in Cognitive Radio Networks: A Queuing-based Approach. *Int. J. Commun. Antenna Propag. IRCEAP* **2017**, *7*, 31–47. [[CrossRef](#)]
18. Xiao, X.; Zeng, F.; Hu, Z. Dynamic Flow-Adaptive Spectrum Leasing with Channel Aggregation in Cognitive Radio Networks. *Sensors* **2020**, *20*, 3800. [[CrossRef](#)] [[PubMed](#)]
19. El-Toukhy, A.T.; Arslan, H. Enhancing the Performance of low-priority SUs Using Reserved Channels in CRN. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 513–517. [[CrossRef](#)]
20. Bayrakdar, M.E.; Calhan, A. Improving Spectrum Handoff Utilization for Prioritized Cognitive Radio Users by Exploiting Channel Bonding with Starvation Mitigation. *Int. J. Electron. Commun.* **2017**, *71*, 181–191. [[CrossRef](#)]
21. Suganthi, N.; Meenakshi, S. An Efficient Scheduling Algorithm Using Queuing System to Minimize Starvation of Non-real-time Secondary Users in Cognitive Radio Network. *Clust. Comput.* **2018**, *25*, 1–12. [[CrossRef](#)]
22. Iftikhar, M.; Mathkour, H.; Imran, M. A Novel Framework for G/M/1 Queuing System based on Scheduling-cum-polling Mechanism to Analyze Multiple Classes of Self-similar and LRD Traffic. *Wirel. Netw.* **2016**, *22*, 1269–1284. [[CrossRef](#)]

23. Khan, M.; Ahmad, J.; Saddik, A.E. Drone-HAT: Hybrid Attention Transformer for Complex Action Recognition in Drone Surveillance Videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Seattle, WA, USA, 19–21 June 2024; pp. 4713–4722.
24. Taherkhani, N.; Pierre, S. Prioritizing and Scheduling Messages for Congestion Control in Vehicular Ad Hoc Networks. *Comput. Netw.* **2016**, *108*, 15–28. [[CrossRef](#)]
25. Han, W.; Li, J.; Tian, Z. Dynamic Sensing Strategies for Efficient Spectrum Utilization in Cognitive Radio Networks. *IEEE Trans. Wirel. Commun.* **2011**, *10*, 3644–3655. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Deep Reinforcement Learning-Based Multipath Routing for LEO Megaconstellation Networks

Chi Han ¹, Wei Xiong ^{1,2,*} and Ronghuan Yu ^{1,2}¹ National Key Laboratory of Space Target Awareness, Space Engineering University, Beijing 101400, China; hanchi@hgd.edu.cn (C.H.); yrh1983@163.com (R.Y.)² School of Space Information, Space Engineering University, Beijing 101400, China

* Correspondence: xiongwei@hgd.edu.cn

Abstract: The expansion of megaconstellation networks (MCNs) represents a promising solution for achieving global Internet coverage. To meet the growing demand for satellite services, multipath routing allows the simultaneous establishment of multiple transmission paths, enabling the transmission of flows in parallel. Nevertheless, the mobility of satellites and time-varying link states presents a challenge for the discovery of optimal paths and traffic scheduling in multipath routing. Given the inflexibility of traditional static deep reinforcement learning (DRL)-based routing algorithms in dealing with time-varying constellation topologies, DRL-based multipath routing (DMR) enabled by a graph neural network (GNN) is proposed as a means of enhancing the transmission performance of MCNs. DMR decouples the stochastic optimization problem of multipath routing under traffic and bandwidth constraints into two subproblems: multipath routing discovery and multipath traffic scheduling. Firstly, the minimum hop count-based multipath route discovery algorithm (MHMRD) is proposed for the computation of multiple available paths between all source and destination nodes. Secondly, the GNN-based multipath traffic scheduling scheme (GMTS) is proposed as a means of dynamically scheduling the traffic on each available path for each data stream, based on the state information of ISLs and traffic demand. Simulation results demonstrate that the proposed scheme can be scaled to constellations with different configurations without the necessity for repeated training and enhance the throughput, completion ratio, and delay by 42.64%, 17.39%, and 3.66% in comparison with the shortest path first algorithm (SPF), respectively.



Citation: Han, C.; Xiong, W.; Yu, R. Deep Reinforcement Learning-Based Multipath Routing for LEO Megaconstellation Networks.

Electronics **2024**, *13*, 3054. <https://doi.org/10.3390/electronics13153054>

Academic Editor: Stefano Scanzio

Received: 12 July 2024

Revised: 25 July 2024

Accepted: 31 July 2024

Published: 1 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: satellite network; multipath routing; deep reinforcement learning; traffic scheduling; hop count

1. Introduction

The characteristics of megaconstellation networks (MCNs), such as wide area coverage, low latency, and high bandwidth, confer upon them unique advantages in remote area communications and delay-sensitive services (e.g., emergency communications and disaster monitoring) [1]. The aforementioned advantages have resulted in the accelerated development of MCNs, which are anticipated to fulfill a multitude of services, including communications, remote sensing, navigation, and positioning. Furthermore, MCNs are poised to become a pivotal component of the infrastructure of the space-air-ground integrated network (SAGIN) [2–4].

In order to meet the growing demand for satellite communication services, multipath routing is a promising approach. Multipath routing allows multiple transmission paths to be established simultaneously between the source and destination nodes for each data stream, which can increase network throughput [5]. In contrast to single-path routing, multipath routing allows for the optimal utilization of available bandwidth, thereby alleviating congestion, improving throughput, and enhancing fault tolerance. This is achieved by enabling a single service to be transmitted over multiple paths in parallel. Consequently,

multipath routing facilitates the resilience of resource-constrained satellite networks to withstand node or link failure scenarios. However, the high dynamics of satellite network topologies and transmission link states presents challenges for multipath routing discovery and multipath traffic scheduling [6–8].

Multipath routing discovery is the first challenge posed by the time-varying nature of satellite topology and traffic. Factors such as changes in satellite attitude and complex electromagnetic environments can cause laser link transmission rates to decrease or even fail. As a result, link failures may occur frequently in MCNs rather than being an exceptional event, as in terrestrial networks. The frequent occurrence of link passes or failures impedes the ability of the broadcast mechanism to discover and maintain routes in a timely manner, as is the case in terrestrial networks. This results in an increase in the overheads associated with broadcast and route control. Consequently, topology-based ad hoc routing protocols, such as ADOV [9], are suboptimal in MCNs. Furthermore, due to the mobility and energy constraints of LEO satellites, established links are often not stably maintained. Topological changes in the path discovery process may result in the rapid expiration of discovered paths, with the corresponding paths no longer existing at the time of data transfer. The above factors require multipath routing discovery to be able to sense link failures in a timely manner and reasonably set multiple alternative paths. Some schemes currently deploy path computation in the ground network control center (NCC) [10]. The NCC improves network transmission performance by comprehensively considering the distribution of terrestrial user traffic and link utilization and making full use of the link redundancy brought about by the mesh topology of the satellite network to plan multiple parallel transmission paths for each flow at the same time.

Multipath traffic scheduling in MCNs is the second challenge. Complex electromagnetic environments, satellite attitude changes, and other factors result in fluctuations in link quality. As a consequence, traditional multipath traffic scheduling based on static routing is no longer applicable in MCNs. Furthermore, the time-varying nature of terrestrial user traffic and the fluctuation in link quality render the satellite network incapable of accurately identifying traffic patterns, and fine-grained traffic scheduling between multiple paths frequently lacks effective information. Deep reinforcement learning (DRL) has demonstrated significant advantages in the description of spatiotemporal features, flow decisions, and other related areas. A considerable number of scholars have opted to adopt DRL for the purpose of adaptive multipath flow scheduling, as evidenced by [11–14]. However, the time-varying satellite networks and fluctuating quality of inter-satellite links (ISLs) necessitate the aggregation of multiple time slices for decision making in models employing DRL. As the constellation size increases, the model's dimension expands dramatically, resulting in a significant increase in computational effort and a reduction in model scalability. Graph neural networks (GNNs) are capable of identifying features and patterns in graph-structured data, performing relational reasoning and combinatorial generalization on graph-structured data [15,16], and they can be applied without additional model tuning to constellations of different topologies and sizes [17,18].

Recently, DRL has demonstrated considerable potential in learning temporal and spatial traffic characteristics for the purpose of making routing decisions. Researchers have employed DRL for the purpose of adaptive traffic assignment, with the aim of replacing static traffic segmentation. However, DRL-based models require the ability to summarize continuous topological snapshots in order to make decisions in a dynamic satellite environment. Satellite networks have high-dimensional and sparse state spaces, and the dimensionality of DRL models increases dramatically with the size of the constellation and the level of traffic demand, which in turn leads to poor model scalability. Graph neural networks (GNN) facilitate the inference of relational data and the generalization of combinatorial structures within graph-structured information. This enables application of the DRL model to satellite constellations of varying sizes without the need for additional modifications. In light of the aforementioned analysis, we propose DRL-based multipath routing (DMR) embedded by a GNN to improve the transmission performance of MCNs.

The multipath routing problem is initially modeled as a stochastic optimization problem, with the objective of maximising network efficiency under traffic and bandwidth constraints. In order to obtain the optimal solution, the original problem is decoupled into two subproblems: multipath routing discovery and multipath traffic scheduling. With regard to the initial subproblem, it is proposed that the redundant topology of the satellite network be fully exploited, with the introduction of a minimum hop count-based multipath routing discovery (MHMRD) algorithm. MHMRD is designed to compute a number of available paths between all source and destination nodes. On this basis, the second subproblem is addressed by the design of GNN-based multipath traffic scheduling (GMTS). GMTS models the multipath traffic scheduling process as a Markov decision process (MDP), which dynamically schedules traffic on each available path for each data stream based on satellite network link state information and traffic demand. Simulation results on Iridium and OneWeb demonstrate that the proposed scheme is not reliant on retraining for different-sized constellations. Furthermore, the network throughput, average delay, and flow completion rate exceeded those of the baseline scheme.

The main contributions of this paper are summarized as follows:

- The multipath routing problem in satellite networks is modeled as a stochastic optimization problem under traffic and bandwidth constraints. The objective is to maximize network efficiency. The problem is decoupled into a multipath routing discovery subproblem and a multipath traffic scheduling subproblem.
- In order to address the multipath routing discovery subproblem, the MHMRD algorithm has been proposed as a means of planning multiple minimum hop count paths for each flow between any pair of nodes in the network based on the link state information.
- The multipath traffic scheduling subproblem is modeled as an MDP, with the GMTS scheme proposed to dynamically schedule the proportion of traffic on each available path for each data stream. GMTS is a scalable solution which can be applied to different constellations.

The rest of this paper is organized as follows. Section 2 provides some related work. In Section 3, we provide the system model and problem formulation. In Section 4, DRL-based multipath routing is described. Section 5 provides the simulation and main results. Finally, we summarize the results in Section 6.

2. Related Work

2.1. Multipath Routing

In satellite networks, there are multiple paths with an equal number of hops from end to end. In order to improve the reliability of data transmission and enhance the network load balancing capability, researchers have proposed the use of multipath routing in satellite networks, where each flow is divided into multiple subflows for parallel transmission along different paths. In order to distribute traffic over multiple paths, the traditional equal-cost multipath (ECMP) method performs static traffic splitting based on message information. However, the ECMP approach does not consider the limitations of network parameters such as the bandwidth and delay and is prone to congestion when the network load is high. To achieve dynamic flow management for multipath routing, a software-defined network (SDN)-based satellite network architecture has been proposed for central topology control and traffic control [18–21]. In the SDN-based network architecture, the data plane and control plane are separated, the SDN controller determines the routing based on the network parameters and QoS requirements, and the satellite is responsible for data forwarding. In [22], the authors attempted to integrate network coding and multipath routing in order to enhance the data transmission efficiency of satellite networks. However, the centralized control approach is susceptible to high latency issues. In [23], in order to reduce the control latency and improve the transmission stability, the authors employed a distributed approach for routing control. In [24], the authors proposed network coding-based multipath cooperative routing (NCMCR). In order to optimize the transmission capacity, each flow

in NCMCR is transmitted along multiple disjoint links. However, the aforementioned multipath routing schemes do not consider the link parameters of the various forwarding paths and thus are unable to adaptively perform reasonable traffic splitting. In [1], the authors designed an adaptive traffic balancing scheme among multiple temporal paths by combining sparse and redundant network coding mechanisms. This approach enables the achievement of deterministic delay guarantees in limited node resources and multi-user competition scenarios. Nevertheless, all of the aforementioned routing schemes utilize static topologies, which precludes them from adapting to satellite networks with dynamically changing topologies.

2.2. Intelligent Routing

The time-varying satellite network environment and dynamic interstellar links bring challenges to satellite network routing. In order to ensure the network's quality of service under the conditions of proliferating user and constellation sizes, multipath routing requires more efficient path planning and finer traffic control strategies. Reinforcement learning is an intelligent tool which supports decision making by interacting with the environment [25–27], and scholars have proposed numerous intelligent routing schemes based on reinforcement learning. In [25], the authors developed a supervised deep learning system for the construction of routing tables. In [28], the authors employed graph-based deep learning in satellite networks through a neural network architecture called the graph-query neural network. Nevertheless, such supervised learning-based routing schemes are constrained by their inability to generalize and adapt across different constellations. To address this issue, the authors of [29] proposed a Markov decision process (MDP) model for satellite network routing, employing multi-agent deep reinforcement learning (DRL) to satisfy diverse quality of service (QoS) requirements. However, the fixed goal policy renders DRL-based routing algorithms inflexible, as they are unable to adapt to dynamically changing networks. It is evident that traditional static objective configurations are unable to reflect the varying importance of different metrics in dynamic network environments. In order to address this issue, the authors of [17,30] integrated a graph neural network (GNN) into DRL intelligences, enabling the timely adjustment of network performance metrics by predicting the trend of the optimization objectives of routing algorithms. This enables the intelligences to learn optimal paths which can adapt to different environmental changes. In [31,32], the authors proposed different DRL architectures, all of which dynamically adjust the link weights according to the load of the key nodes or the link load. In [33], the authors extended the TCP options to piggyback the relevant control information and flexibly support communication between the subflows of the transport layer and the SDN controller. An SDN cooperated MPTCP (scMPTCP) architecture was proposed, which selects routes for new subflows based on the available bandwidth of each route and avoids the bottleneck of other subflows. It also can adapt to changes in network load. In [34], the authors proposed a DQN controller considering a path loss model based on the Markov decision process model for network selection and adaptive resource allocation in heterogeneous networks.

Nevertheless, an examination of the aforementioned research reveals that existing DRL-based studies are not readily scalable due to their inherent coupling to the input states and limited topology, which is consistent with the training data. To resolve the aforementioned contradiction, the proposed DRL-based multipath routing embedded with a GNN can readily identify the optimal action within a continuous action space. DMR decouples the multipath routing problem into distinct components—multipath discovery and multipath traffic scheduling—to reduce the solution's complexity. Concurrently, the target policy adopted by DMR can be modified in accordance with the prevailing environmental conditions, thereby enhancing the flexibility of the system with respect to different topologies.

3. System Model and Problem Formulation

3.1. Multipath Scenario

The multipath routing scenario in the MCN is shown in Figure 1, where the satellite network is divided into mutually independent data planes and control planes via the SDN. In the terrestrial network, data in the send buffer are transmitted to the sender through routers over multiple hops and aggregated into multiple subflows. Each subflow can select the optimal transmission path according to the current network condition and then converge at the receiver.

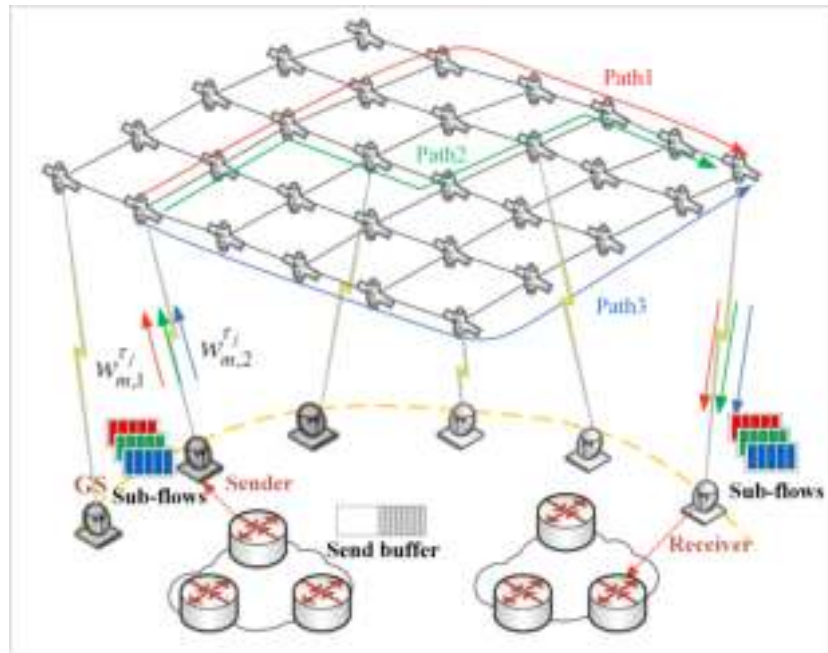


Figure 1. Multipath routing in the megaconstellation network.

The data plane comprises the Walker Delta constellation $N_S N_P / N_P / F$, which is distributed in a uniform and symmetrical manner, and the ground station. The N_P orbital planes are distributed uniformly along the equator, while the N_S satellites are distributed uniformly on each plane. Each satellite is capable of being connected to four inter-satellite links (ISLs), comprising two intra-plane ISLs and two inter-plane ISLs. In the control plane, the network control center assumes the functions of unified topology management, traffic scheduling, and path control. Due to the mobility of LEO megaconstellation networks, inter-plane ISLs are subject to frequent disruptions and rebuilds, and the topology is in a state of constant flux. Consequently, the NCC must be continuously updated to accommodate the evolving topology, which presents a significant challenge to multipath routing planning and traffic scheduling.

3.2. Multipath Routing and Traffic Model

The MCN is modeled as a spatiotemporal graph $G = \{(V, E, T, N)\}$, where $V = S \cup G$ is the node consisting of all satellites and terrestrial gateways and $E = E_S \cup E_G$ is the set of ISLs E_S and the set of terrestrial links E_G , while T is the time slot vector of a length N . As illustrated in Figure 1, the mesh topology of the MCN allows for the existence of multiple potential routes between any given pair of nodes. Assume that the number of packet flows at time slot t is M (i.e., $P_t = \{(s_i, d_i) | i = 1, 2, \dots, M\}$). In other words, the variable M represents the number of subflows, as shown in Figure 1, where three subflows in the sender are awaiting transmission via multiple paths. The number of available paths between the source node s_i and destination node d_i is L_{s_i, d_i} , denoted as $R_t = \{p_{1,1}^t, \dots, p_{1,L}^t, \dots, p_{M,L}^t\}$. Here, $p_{m,l}^t$ is the l th path between the m th pair of source and destination nodes, consisting

of all ISLs on that path (i.e., $p_{m,l}^t = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_j, \dots, \varepsilon_{n_{m,l}}\}$, where $n_{m,l}$ is the total hop count of the current link).

Let $q_{\varepsilon_j}^t$ be the queuing delay of the link ε_j , $r_{\varepsilon_j}^t$ be the packet processing delay, and $\gamma_{\varepsilon_j}^t$ be the link distance. Then, the delay of the l th transmission path of the m th source-destination node pair can be expressed as follows:

$$\eta_{m,x}^t = \sum_{j=1}^{n_{m,x}} \left(q_{\varepsilon_j}^t + r_{\varepsilon_j}^t + \frac{\gamma_{\varepsilon_j}^t}{c} \right) \quad (1)$$

where c is the vacuum speed of light and $n_{m,l}$ denotes the hop count of the current link. Each source node can divide the traffic into L_{s_i,d_i} subflows to pass through L_{s_i,d_i} paths in parallel. Let $\omega_{m,l}^t$ be the traffic ratio of each subflow. Then, the sum of L_{s_i,d_i} subflows between any pair of source and destination nodes should be one, which can be expressed as follows:

$$\sum_{l=1}^L \omega_{m,l}^t = 1, \forall m = 1, 2, \dots, M \quad (2)$$

Therefore, the traffic of link ε_j at time t can be expressed by

$$f_{\varepsilon_j}^t = \sum_{m=1}^M \sum_{l=1}^L b_m^t \omega_{m,l}^t \delta_{m,l}^t \quad (3)$$

where b_m^t is the bandwidth requirement of the m th pair to the source and destination nodes and $\delta_{m,l}^t$ is the path packet loss rate. Noting that the maximum bandwidth of the interstellar link is BW , the ISL traffic should not exceed the maximum bandwidth limit, which can be expressed as follows:

$$f_{\varepsilon_j}^t \leq BW, \forall \varepsilon_j \in p_{m,l}^t \quad (4)$$

Consequently, the throughput between the m th node pair can be expressed by

$$f_{m,l}^t = \max \{ f_{\varepsilon_j}^t \}, \forall \varepsilon_j \in p_{m,l}^t \quad (5)$$

3.3. Problem Formulation

In order to measure the delay and throughput of ISLs, the utility function of the MCN is defined as

$$U(\bar{f}_t, \bar{d}_t) = \beta_1 \log(\bar{f}_t) - \beta_2 \log(\bar{d}_t) \quad (6)$$

where β_1 and β_2 are the importance coefficients of the throughput and delay, respectively, β_1 and β_2 satisfy $\beta_1 + \beta_2 = 1$, and $\bar{f}_t$ and $\bar{d}_t$ are the average throughput and delay of the current time slot, respectively, which can be expressed as follows:

$$\bar{f}_t = \sum_{m=1}^M \sum_{l=1}^{|f_m|} \sum_{y=1}^{|p_k|} \frac{\nabla_{pak}}{t}, \forall t \in [1, T] \quad (7)$$

$$\bar{d}_t = \frac{\sum_{m=1}^M \sum_{l=1}^{|f_m|} \sum_{y=1}^{|p_k|} d_{m,l}^t \kappa_y}{\sum_{m=1}^M \sum_{l=1}^{|f_m|} \sum_{y=1}^{|p_k|} \kappa_y}, \forall t \in [1, T] \quad (8)$$

where $|f_m|$ is the number of data flows between the m th node pair, $|p_k|$ is the number of packets for the k th data flow, ∇_{pak} is the packet size, and $\kappa_y \in \{0, 1\}$ is a binary variable which measures whether the y th packet was successfully sent.

Consequently, the satellite network multipath routing optimization can be expressed as the following optimization problem:

$$\begin{aligned}
 P0 : & \max U(\bar{f}, \bar{d}), \forall t \in [1, T] \\
 C1 : & \bar{d} = \frac{1}{N} \sum_{i=1}^N \bar{d}_t \\
 C2 : & \bar{f} = \frac{1}{N} \sum_{i=1}^N \bar{f}_t \\
 s.t. \quad C3 : & f_{\varepsilon_j}^t \leq BW, \forall \varepsilon_j \in p_{m,l}^t \\
 C4 : & \sum_{l=1}^L \omega_{m,l}^t = 1, \forall m \in [1, M] \\
 C5 : & d_{m,l}^t \leq d_{m,y}^t, \forall 1 \leq l \leq y \leq L, \forall m \in [1, M]
 \end{aligned} \tag{9}$$

where constraints C1 and C2 denote the average delay and throughput of the satellite network, respectively, constraint C3 stipulates that the throughput of link ε_j at any given moment t cannot exceed the bandwidth limit, constraint C4 denotes that the total communication between any pair of nodes is the cumulative result of multiple sub-streams traversing multiple paths, and finally, constraint C5 describes the delay constraints of all candidate paths between each pair of nodes. It can be seen that P0 is an NP-hard problem with a rugged solution space. P0 consists of two parts—multipath routing discovery and multipath traffic scheduling—which converge slowly and may fall into local optima if solved by stochastic optimization. In order to reduce the solution's complexity, this paper decouples the problem into two subproblems, routing discovery and traffic scheduling, to improve the efficiency of the algorithm.

4. DRL-Based Multipath Routing

In this paper, we propose DRL-based multipath routing (DMR), as shown in Figure 2, which first uses the minimum hop count based-multipath discovery algorithm to solve for multiple paths with acceptable delays and then utilizes the GMTS algorithm to determine the traffic scheduling scheme based on the state of the constellation network.

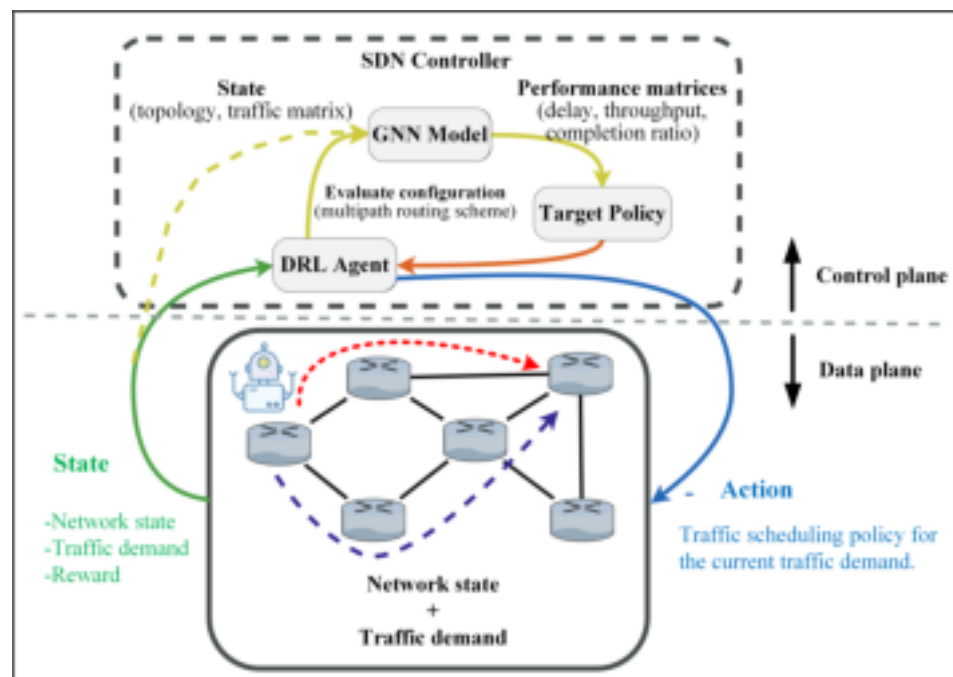


Figure 2. The overall structure of DRL-based multipath routing.

4.1. Multipath Routing Discovery

The objective of the multipath routing discovery subproblem is to identify the optimal delay path and the suboptimal delay path. In MCNs, the propagation delay, which depends on the number of path hops, represents a significant component of the end-to-end delay. Consequently, multipath paths are primarily considered for the hop-optimal and suboptimal paths within the network. The multipath routing discovery subproblem can be expressed as follows:

$$\begin{aligned} P1 : \min & d_{m,l}^t, \forall t \in T, m \in [1, M], l \in [1, L] \\ \text{s.t.} \quad & d_{m,l}^t \leq d_{m,y}^t, \forall 1 \leq l \leq y \leq L \end{aligned} \quad (10)$$

The minimum hop count path between any pair of nodes is initially determined. On this basis, the NCC determines the suboptimal transmission path (i.e., the backup path) for each flow in descending order of traffic. This is because centralized traffic is more likely to cause bottleneck link congestion than decentralized tiny traffic. In the Walker Delta constellation $N_S N_P / N_P / F$, as shown in Figure 3, the inclination is α , and the right ascension of the ascending node (RAAN) difference of adjacent orbits is $\Delta\Omega = 2\pi / N_P$. The phase difference between neighboring satellites in the same orbit is $\Delta\Phi = 2\pi / N_S$, while the phase difference between neighboring satellites in adjacent orbits is $\Delta v = 2\pi F / N_S N_P$, while u_1 and u_2 are the argument of latitude (or phase angle) values of the source and destination satellites, respectively. The argument of latitude is the angle between the ascending node and the satellite and basically defines the position of the satellite in the orbit, where $\zeta(u_1)$ and $\zeta(u_2)$ indicate the longitude difference of the source and destination satellite to the corresponding ascending nodes, respectively. In the Walker Delta constellation, the end-to-end hop count includes both transverse inter-plane hops H_h and intra-plane hops H_v .

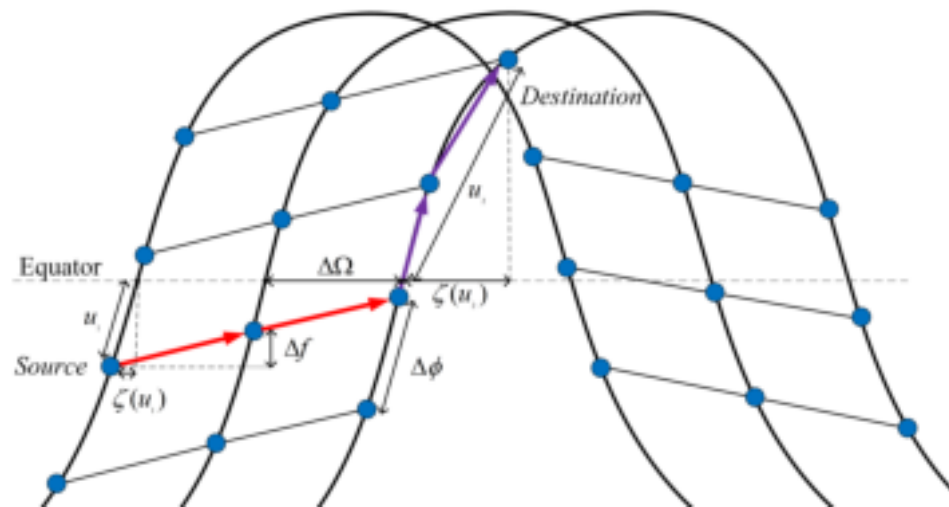


Figure 3. Track of sub-satellite point of megaconstellation networks.

4.1.1. Inter-Plane Hops H_h

The number of inter-plane hops depends on the RAAN difference ΔY_0 of the orbits where the source and destination nodes are located. Given a source and destination satellite pair (s, d) , ΔY_0 is calculated as follows:

$$\Delta Y_0 = (Y_2 - Y_1) \bmod 2\pi \in [0, 2\pi] \quad (11)$$

where Y_1 and Y_2 are the RAAN values of s and d , respectively. If the destination satellite is to the west of the source satellite, then the RAAN difference is $2\pi - \Delta Y_0$. Since the RAAN

difference between adjacent orbits is constant (i.e., $\Delta\Omega = 2\pi/N_p$), the inter-plane hops in the west and east directions can be expressed as follows:

$$H_h^{\leftarrow} = \left\langle \frac{2\pi - \Delta Y_0}{\Delta\Omega} \right\rangle \quad (12)$$

$$H_h^{\rightarrow} = \left\langle \frac{\Delta Y_0}{\Delta\Omega} \right\rangle \quad (13)$$

where $\langle x \rangle = \text{sgn}(x) \lfloor x + 1/2 \rfloor$ denotes the integer closest to x , $H_h^{\leftarrow}$ is the hop count between the orbital planes to the west, and $H_h^{\rightarrow}$ is the eastward inter-plane hop count.

4.1.2. Intra-Plane Hops H_v

The intra-plane hops depend on the phase angle difference Δu of the satellite. Each intra-plane hop increases the phase angle by $\Delta\Phi$, and each inter-plane hop results in a phase angle increment of Δf . Therefore, the phase angle of the destination satellite can be expressed as follows:

$$u_2 = u_1 + \Delta f \cdot H_h^{\rightarrow} + \underbrace{\Delta\Phi \cdot H_v^{\nearrow}}_{\Delta\vec{u}} \quad (14)$$

where $H_v^{\nearrow}$ is the eastward intra-plane hop count. In order to compute H_v , it is necessary to first eliminate the phase difference $\Delta f \cdot H_h^{\rightarrow}$ caused by H_h from the phase difference $\Delta u = u_2 - u_1$, where u_1 and u_2 are the argument of latitude values of the source and destination satellites, respectively. To distinguish between the phase difference due to eastward and westward propagation, Δu can be expressed separately as follows:

$$\Delta\vec{u} = (u_2 - u_1 - H_h^{\rightarrow} \cdot \Delta f) \bmod 2\pi \quad (15)$$

$$\Delta\overleftarrow{u} = (u_2 - u_1 + H_h^{\leftarrow} \cdot \Delta f) \bmod 2\pi \quad (16)$$

where Δf is the phase angle change due to inter-plane hops. As illustrated in Figure 3, the track of the sub-satellite point is divided into an ascending segment (from southwest to northeast) and a descending segment (from northwest to southeast), allowing for the propagation of data in both directions. Consequently, in this paper, the intra-plane hops in the four directions are calculated, which can be expressed as follows:

$$H_v^{\nwarrow} = \left\lceil \frac{\Delta\overleftarrow{u}}{\Delta\Phi} \right\rceil \quad (17)$$

$$H_v^{\nearrow} = \left\lceil \frac{\Delta\vec{u}}{\Delta\Phi} \right\rceil \quad (18)$$

$$H_v^{\swarrow} = \left\lceil \frac{2\pi - \Delta\overleftarrow{u}}{\Delta\Phi} \right\rceil \quad (19)$$

$$H_v^{\searrow} = \left\lceil \frac{2\pi - \Delta\vec{u}}{\Delta\Phi} \right\rceil \quad (20)$$

where $\Delta\overleftarrow{u}$ and $\Delta\vec{u}$ denote the phase difference due to westward and eastward propagation between adjacent orbits, respectively. Meanwhile, $H_v^{\nwarrow}$, $H_v^{\nearrow}$, $H_v^{\swarrow}$, and $H_v^{\searrow}$ denote the phase difference due to hops in the orbit toward the northwest, northeast, southwest, and southeast, respectively.

The inter-plane hops H_h and the intra-plane hops, H_v were calculated earlier. Consequently, the end-to-end minimum hop count in the inclined orbit constellation can be expressed as follows:

$$H = \min \left\{ \begin{array}{l} H_h^{\leftarrow} + H_v^{\nearrow} \\ H_h^{\leftarrow} + H_v^{\searrow} \\ H_h^{\rightarrow} + H_v^{\nearrow} \\ H_h^{\rightarrow} + H_v^{\searrow} \end{array} \right\} \quad (21)$$

On the basis of obtaining the shortest hop count path, multiple available paths are generated. Taking the suboptimal path computation between node pairs (s, d) as an example, firstly, the occupancy frequency $F_{a,b}^t$ of each link and the minimum hop count path set $R_{a,b}^t$ are computed. Meanwhile, the link occupancy frequency threshold ζ_ϵ is defined. The weight $m_{a,b}^t \in [r_1, r_2]$ of link (a, b) is set if $F_{a,b}^t \leq \zeta_\epsilon$; otherwise, $m_{a,b}^t \in [r_2, r_3]$, where $r_1 < r_2 < r_3$ are random numbers. In order to prevent the interconnection of disparate paths, the NCC will eliminate the links in $R_{a,b}^t$ from the network. Subsequently, the NCC will identify the optimal feasible paths for (s, d) among the remaining links, utilising the principle of the shortest end-to-end hop count. The aforementioned process is repeated continuously until L possible paths between any two nodes in the network have been identified. The specific process of MHMRD is illustrated in Algorithm 1. The computational complexity of Algorithm 1 is $O(N \log N)$.

Algorithm 1: Minimum hop count-based multipath route discovery (MHMRD).

Input: $\{(V, E, T, N)\}$, source-destination node pair P

Output: Candidate paths $\mathbf{R}$

```

1 Initialization:  $\mathbf{R} = \emptyset$ 
2 for  $i = 1 : N$  do
3   Compute the minimum hop count path  $\mathbf{R}^t$  via Equation (21);
4   Compute other available paths;
5   for  $(s, d)$  in  $P$  do
6     for  $l = 1 : L - 1$  do
7       Compute the occupancy frequency  $F_{a,b}^t$  of each ISL;
8       for  $(a, b)$  in  $\mathbf{R}^t$  do
9          $F_{a,b}^t \leftarrow F_{a,b}^t + 1$ ;
10      end
11      for  $(a, b)$  in  $\mathbf{E}^t$  do
12        if  $F_{a,b}^t \leq \zeta_\epsilon$  then
13           $m_{a,b}^t \in [r_1, r_2]$ ;
14        end
15        if  $F_{a,b}^t > \zeta_\epsilon$  then
16           $m_{a,b}^t \in [r_2, r_3]$ ;
17        end
18      end
19      for  $e$  in  $\mathbf{R}_{s,d}^t$  do
20        Remove the edge from  $\mathbf{E}^t$ ;
21      end
22      Compute the feasible paths  $\mathbf{R}$  via minimum hop count path;
23    end
24  end
25 end
```

4.2. Multipath Traffic Scheduling

Based on the completion of multipath path discovery, the second subproblem of multipath routing for MCNs is multipath traffic scheduling, which can be expressed as follows

$$\begin{aligned}
 P2 : \max U(\bar{f}, \bar{d}), \forall t \in T \\
 \text{s.t.} \quad & \sum_{l=1}^L \omega_{m,l}^t = 1, \forall m = 1, \dots, M \\
 & f_{\varepsilon_j}^t \leq BW, \forall \varepsilon_j \in p_{m,l}^t
 \end{aligned} \quad (22)$$

Since the traffic assignment decision in satellite networks depends on the current node and surrounding node states in the network, independent of the historical state, the multipath traffic scheduling subproblem can be described as a Markov decision process (MDP). The occurrence of failures in ISLs or nodes has the effect of impairing the functionality of the satellite network, resulting in frequent alterations to the network topology. Given that the GNN has a superior generalization capability for topologies of varying sizes, this paper proposes the adoption of GNN-based multipath traffic scheduling (GMTS), as illustrated in Figure 4. The DRL agent uses a proximal policy optimization (PPO) algorithm, which is an actor-critic algorithm. At each stage of the MDP, the actor model selects a traffic scheduling decision, and the critic model assigns a score to that decision based on the environmental reward feedback. The process is repeated continuously, with the objective of optimizing the flow scheduling task in order to maximize the cumulative reward.

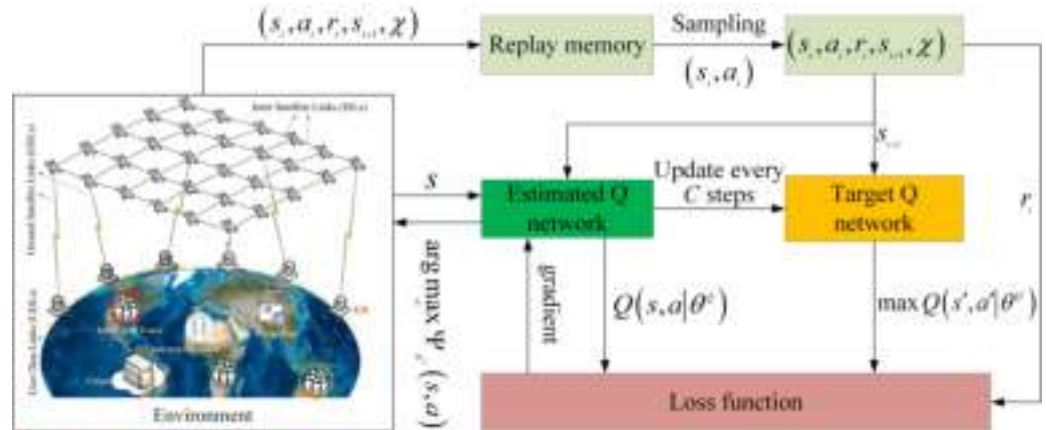


Figure 4. The framework of the proposed GNN-based multipath traffic scheduling (GMTS) algorithm.

4.2.1. State

By dividing the time into multiple time slots, the traffic demand matrix within each time slot can be denoted as $TR = [b_1, b_2, \dots, b_M]$, where b_i denotes the bandwidth between the i th pair of nodes. According to Equation (5), the throughput of the j th link is f_j . Consequently, the residual bandwidth of the i th link, denoted by c_i , can be expressed in terms of the link bandwidth BW :

$$c_i = BW - f_i, i \in [1, L] \quad (23)$$

Given that the maximum number of hops between pairs of nodes is $H_{\max}$, the j th path between the i th pair of nodes can be designated as $p_{i,j} = \{e_1, e_2, \dots, e_{H_{\max}}\}$, and the corresponding path matrix can be designated as $P_t = [p_{1,1}, \dots, p_{1,L}, \dots, p_{M,L}]$. Therefore, the state vector at time slot t can be expressed as follows:

$$\mathbf{s}_t = [C_t, TR_t, G_t, P_t], \mathbf{s}_t \in S \quad (24)$$

where S is the satellite network state space.

4.2.2. Action

Based on the information in the state space, NCC splits the network traffic between each node pair onto L candidate paths which are precomputed. Therefore, the action is defined as a vector $\mathbf{a}_t$ consisting of the proportion of traffic on the candidate paths, which can be expressed as follows:

$$\mathbf{a}_t = [\omega_{1,1}^t, \dots, \omega_{1,L}^t, \dots, \omega_{M,L}^t] \quad (25)$$

where the proportion of flow on each path should satisfy Equation (2).

4.2.3. Reward

After executing action $\mathbf{a}_t$ under the state $\mathbf{s}_t$ at time t , the agent will gain feedback to evaluate the effectiveness of the action. In multipath traffic scheduling, the reward is the objective function of subproblem P2, which can be expressed as follows:

$$r_t = \sum_{m=1}^M \sum_{l=1}^L U(\bar{f}_{m,l}, \bar{d}_{m,l}) \quad (26)$$

where $\bar{f}_{m,l}$ is the average throughput on the l th path between the m th pair of nodes and $\bar{d}_{m,l}$ is the average delay on the l th path between the m th pair of nodes. The agent will seek to complete the multipath traffic scheduling with a higher throughput and lower latency. Since the decision making is divided into multiple phases, the cumulative reward R_t , based on a defined single momentary reward, can be expressed as follows:

$$R_t = \sum_{k=0}^{\infty} \rho^k r_{t+k+1} \quad (27)$$

where $\rho \in [0, 1]$ is the discount factor. Given that there are multiple possible actions in each state, the average gain for each state and the average gain for each state action can be expressed as follows:

$$\Gamma_{\mu}(s) = E_{\mu}(G_t | \mathbf{s}_t = s) \quad (28)$$

$$\Psi_{\mu}(s, a) = E_{\mu}(G_t | \mathbf{s}_t = s, \mathbf{a}_t = a) \quad (29)$$

Here, $\Psi_{\mu}(s, a)$ is maximized when the agent of the DRL finds the optimal policy μ^* , converting the process of updating $\Psi_{\mu}(s, a)$ with the number of iterations to the Bellman form; in other words, we have

$$\Psi_{\mu}(s, a) = E_{\mu}[r_t + \rho \Psi_{\mu}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) | \mathbf{s}_t = s, \mathbf{a}_t = a] \quad (30)$$

After the DRL converges to a stable level, the corresponding optimal action a_{μ^*} can be expressed as follows:

$$a_{\mu^*} = \arg \max_a \Psi_{\mu^*}(s, a) \quad (31)$$

4.3. Training Process of GMTS

The pseudo-code for the GMTS training process is shown in Algorithm 2 with the computational complexity of $O(L)$. The initialization of $\mu(s|\theta^{\mu})$ and $Q(s, a|\theta^Q)$ is first required, where θ^{μ} and θ^Q are hyperparameters of the actor network and critical network, respectively. During the training phase, the experience replay buffer is employed to store environmental interaction data, including states, actions, and rewards. In order to prevent overfitting, the generated samples $(s_t, a_t, r_t, s_{t+1}, \chi)$ are stored in the experience replay buffer at each decision step. At this point, $\chi \in \{0, 1\}$ is used to determine whether the training of the model is complete in the current phase. The sample $(s_i, a_i, r_i, s_{i+1}, \chi)$ will be extracted at specified intervals in

accordance with the playback buffer threshold $B_{\max}$ with the objective of training the network. At this point, the target Q value can be expressed as follows:

$$y_i = r_i + \rho(1 - \chi) \left[Q(s_{i+1}, a_{i+1} | \theta^Q) \right] \quad (32)$$

Algorithm 2: Training process of the GNN-based multipath traffic scheduling (GMTS).

Input: $B_{\max}$, critic network, actor network, and target network

Output: a_{μ^*}

```

1 for  $i = 1 : L$  do
2   Reset the initial state as  $s_0$ 
3   for  $t = 1 : T$  do
4     Compute the action  $a_t$ , reward  $r_t$ , and new state  $s_{t+1}$ 
5     Store  $(s_t, a_t, r_t, s_{t+1}, \chi)$  in the replay buffer
6     Update state  $s_t$  to  $s_{t+1}$  while  $|B| \leq B_{\max}$  do
7       for  $j = 1 : N_B$  do
8         Calculate  $p_i$  according to Equation (34)
9         Calculate  $\omega_i$  according to Equation (36)
10        Calculate the target Q value  $y_i$  according to Equation (32)
11        Update the probability of sampling transition according to
            Equation (35)
12      end
13    end
14    Calculate the loss of the critic network and actor network according to
        Equations (38) and (39)
15    Update the parameter of actor network and critic network based on the
        gradient descending approach.
16  end
17 end

```

In the GMTS algorithm, the temporal difference error (TD error) is employed as a means of gauging the significance of transitions. The TD error between the target Q value and the current Q value can be expressed as follows:

$$\delta_i = \left| y_i - Q(s_i, a_i | \theta^Q) \right| \quad (33)$$

The corresponding proportional prioritization is

$$p_i = \delta_i + \eta \quad (34)$$

where η denotes an extremely small amount and η is added so that the samples can be sampled even when the TD error is zero. The stochastic sampling method is adopted, and the probability of sampling transition i is

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha} \quad (35)$$

where α determines how much prioritization is used. When $\alpha = 0$, then the sampling method is degraded to random sampling. The importance sample (IS) weights of $(s_i, a_i, r_i, s_{i+1}, \chi)$ are

$$\omega_i = \left(\frac{P(i)}{P_{\min}} \right)^\beta \quad (36)$$

where β is used for biased and unbiased control. Given the learning rate θ^Q , the actor network θ^Q can be updated in the following way:

$$\theta^Q \leftarrow \theta^Q + \theta^Q \delta_i \nabla_{\theta^Q} Q(s, a|\theta^Q) \quad (37)$$

Therefore, the loss function of the critic network and actor network can be expressed as follows:

$$L_q = \frac{1}{N_B} \sum_{i=1}^{B_{\max}} \omega_i \delta_i^2 \quad (38)$$

$$L_\mu = -Q(s, a|\theta^Q) \quad (39)$$

where N_B is the size of the replay buffer B .

4.4. Workflow of DMR

The DRL-based multipath routing (DMR) proposed in this paper can be divided into two phases, minimum hop count-based multipath path discovery (MHMRD) and GNN-based multipath traffic scheduling (GMTS), as shown in Figure 5. In the first stage, the network control center first plans multiple optimal paths between arbitrary node pairs based on the minimum hop count principle according to the network topology. Specifically, based on the periodicity and predictability of the satellite orbits, the network is divided into multiple time slices according to time slot intervals, and the network topology remains stable in each discrete time slice. Within each time slot, MHMRD calculates L minimum hop count available paths for each service based on the priority order defined by the MHMRD based on the end-to-end service volume. After completing the calculation for the current timeslot, it moves to the next timeslot. In the second phase, the network control center generates the split ratios for each service on different paths based on the model obtained from GMTS training and sends them to the corresponding satellites at the appropriate time before the start of each time slot.

On the one hand, this paper adopts priority experience replay to speed up the convergence of the model by sampling the samples in the experience replay pool according to a certain priority. On the other hand, due to the dynamic nature of satellite networks, the network topology and traffic matrices are time-varying, and traditional DNNs cannot effectively handle such changing inputs. Therefore, in this paper, we adopt GNN for variable size graph structures and route configurations accordingly. Since GNNs have better generalization capabilities for dynamic network structures at different scales, they can be used to aggregate elementary features without specifying the input dimensions.

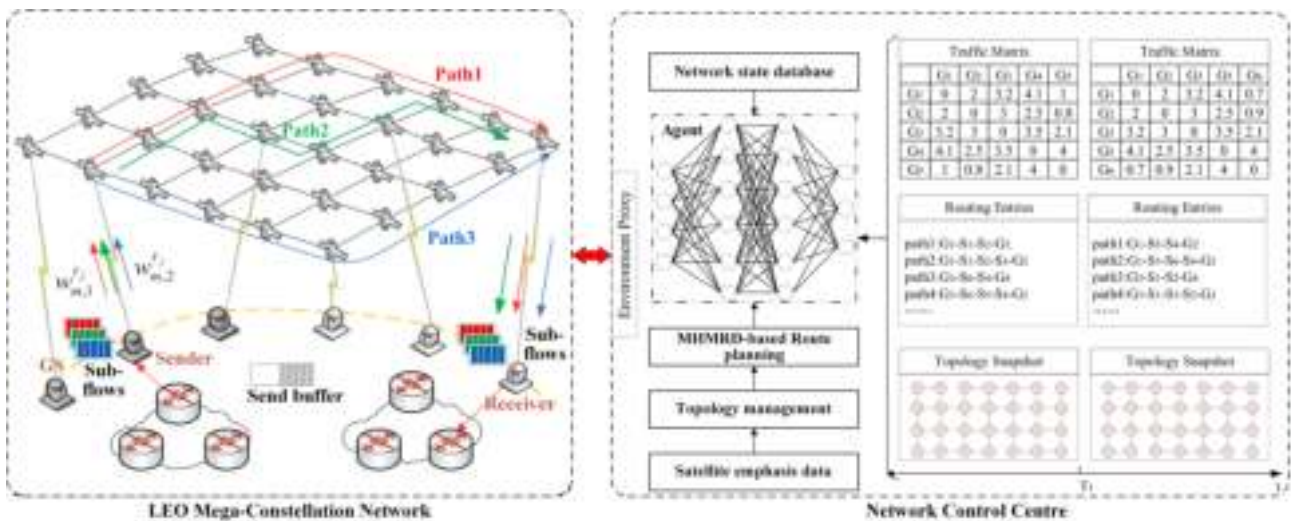


Figure 5. DRL-based multipath routing for LEO megaconstellation networks.

5. Performance Evaluation

5.1. Simulation Set-Up

In this paper, we implement routing computation and multipath traffic scheduling for LEO megaconstellations in NS3 to analyze the performance of the proposed DMR algorithm. In order to analyze the scalability of multipath routing on different constellations, this paper conducts tests on two different-sized constellations. The first constellation was the Iridium constellation [35] with 66 polar orbiting satellites, and the second was the OneWeb constellation [36] with 648 satellites, which contains a total of 18 orbital planes with 36 satellites evenly distributed on each orbital plane. The parameters of the two constellations are shown in Table 1. In order to validate the performance of multipath routing under different sizes of traffic demand, a traffic dataset was generated based on the ground traffic density, where the number of source-destination node pairs was 50. The proposed GMTS algorithm was implemented based on Python 3.9 and Pytorch 1.14, and the corresponding parameters are shown in Table 2.

Table 1. Constellation parameters.

Constellation	N_P	N_S	N_G	ϵ	h	E
Iridium	6	11	16	90°	780 km	20°
OneWeb	18	36	16	53°	550 km	20°

Table 2. Training parameters.

	Parameter	Value
Network parameter	N_g	24
	L_q	100
	∇_{pak}	1 KB
Learning parameter	N_B	32
	B_{max}	100
	α	0.6
	β	0.5
	ρ	0.8

The performance of the proposed minimum hop count multipath path discovery-based (MHMRD) algorithm, which distributes end-to-end traffic in equal proportions to precomputed multiple paths, was tested first. Based on this, the GMTS algorithm was trained using inclined orbit constellation topology and traffic matrices [37], which were deployed to the NCC after training and tested for key metrics such as the average throughput, latency, and flow completion rate of the satellite network. The following comparison algorithms were used in this paper:

- Shortest path first algorithm (SPF): SPF employs the average transmission delay as a link metric with the objective of minimizing the total delay.
- Network coding-based multipath cooperative routing (NCMCR) [24]: NCMCR is designed to address the routing challenges posed by a frequently changing topology and potentially sparse and intermittent connectivity. The NCMCR algorithm takes advantage of the predictability of the relative motion of the satellites, with the time-varying topology modeled as a spatiotemporal map.
- Ant-based multipath backbone routing for load balancing (AMBRLB) [38]: In order to overcome the issues of traffic overflow and overhead, an ant-based load balancing multipath backbone routing algorithm in MANET was proposed. Upon the initiation of transmission by a source node to a destination node, the ant colony optimization (ACO) algorithm is employed to identify multiple paths with the highest probability of success.

- Deep deterministic policy gradient traffic engineering (DDPG-TE) [39]: This method employs the DDPG algorithm to dynamically allocate the traffic proportions of different transmission paths. It should be noted that the model is not applicable to arbitrary satellite constellations, as the dimensions of the state and action spaces are constrained by the size of the input topology and traffic matrices.

5.2. Results and Analysis

5.2.1. Throughput

The average throughput of the MCN over the Iridium constellation and OneWeb constellation are illustrated in Figure 6, with each data point representing the average value of the throughput under five consecutive topological snapshots. It can be observed that, in the context of multipath routing, as the input traffic volume increased, a greater proportion of traffic was allocated to multiple paths for forwarding, based on different policies. This resulted in an overall increase in the average network throughput. Nevertheless, the increase in throughput gradually slowed down. The proposed DMR algorithm exhibited a superior throughput at varying user sizes. DMR enhanced the average throughput by 30.26% and 8.45% in comparison with SPF and NCMCR, respectively, when the total service volume reached its upper limit. Figure 6b illustrates the throughput of multipath routing on the OneWeb constellation. It can be observed that DMR employed the same model as that used for the Iridium constellation, while the other algorithms were retrained based on the characteristics of the new network. It can be observed that the DMR algorithm continued to demonstrate superior performance relative to the other benchmark algorithms in the context of changes in the constellation topology and size, reflecting the enhanced generalization ability of DMR. When the traffic intensity was 8 Gbps, DMR demonstrated an improvement in the average throughput of 42.64% and 9.55% in comparison with SPF and NCMCR, respectively.

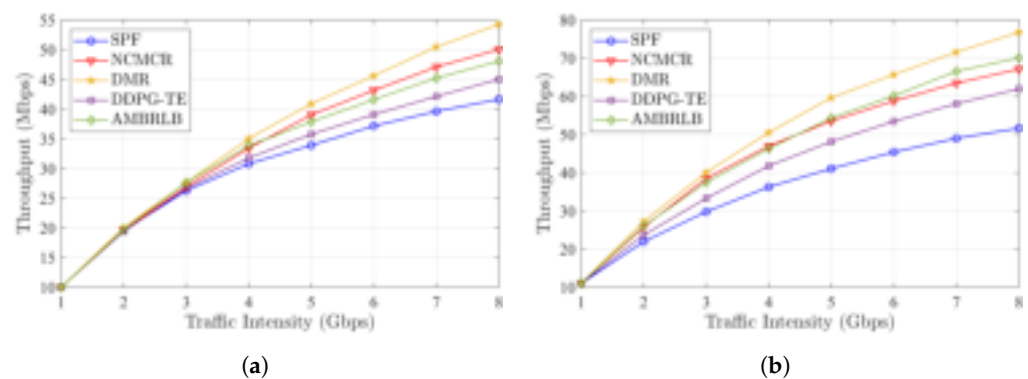


Figure 6. Average throughput on different constellations with various traffic intensity. (a) Average throughput for Iridium. (b) Average throughput for OneWeb.

5.2.2. Average Flow Completion Ratio

Figure 7 illustrates the average flow completion ratio for varying traffic intensities on the Iridium and OneWeb constellations. As the intensity of traffic increased, some nodes in the network gradually became congested. This is evidenced by a decreasing trend in the average flow completion ratios of all five routing schemes. At a traffic intensity of 8 Gbps, the average flow completion ratio of the proposed DMR scheme was enhanced by 17.39% and 11.52% on the Iridium and OneWeb constellations, respectively, in comparison with SPF. Meanwhile, on the OneWeb constellation, the average stream completion rate of the DMR algorithm improved by 1.98%, 2.58%, and 6.49% compared with NCMCR, AMBRLB, and DDPG-TE, respectively. Consequently, it can be seen that the average flow completion ratio on the larger OneWeb constellation was consistently higher than that on the smaller Iridium constellation, even under varying traffic intensities. This demonstrates the advantage of large-scale constellations in handling high loads. It can therefore be concluded that the

proposed DMR scheme demonstrates superior generalization ability across different scale constellations, with a higher flow completion ratio than the baseline scheme.

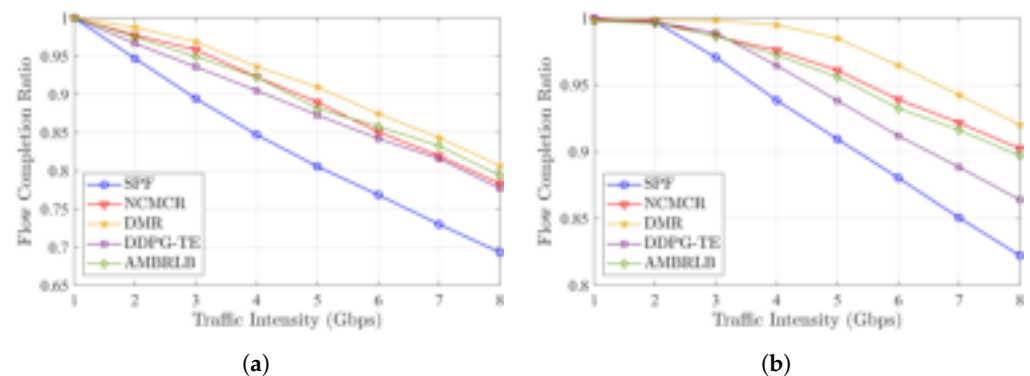


Figure 7. Average flow completion ratio of different constellations with various traffic intensity. (a) Average flow completion ratio for Iridium constellation. (b) Average flow completion ratio for OneWeb constellation.

5.2.3. Average End-to-End Delay

Figure 8 illustrates the average end-to-end delay as a function of the traffic intensity on the Iridium and OneWeb constellations. As the network load increased, the queuing delay also rose, with more flows transmitted along the candidate paths with higher hop counts. Consequently, the average end-to-end delay gradually increased for all of the compared schemes. In the Iridium constellation, SPF achieved the lowest end-to-end delay when the network load was low. This is due to the fact that the scheme always transmits the packets along the shortest path when all the links in the network are in a normal state. As the network load increased, when the total service volume reaches the upper bound, SPF is unable to dynamically change the path to avoid congested paths caused by the increased load. Consequently, on the Iridium constellation, the DMR scheme reduced the average end-to-end delay by 3.67%, 1.78%, 1.86%, and 0.86% compared with SPF, NCMCR, DDPG-TE, and AMBRLB, respectively. Upon extending multipath routing to larger OneWeb constellations, the delay for DMR approached that of NCMCR, AMBRLB, and DDPG-TE when the traffic intensity reached its upper limit. This is due to the fact that both the DMR scheme and the DDPG-TE scheme take into account the dynamic distribution of traffic across different paths. In general, the mean end-to-end delay of DMR at a high load was situated between those of SPF and DDPG-TE. The primary objective of multipath routing is to enhance the network's throughput and resilience when the topology is disrupted. In small-scale constellations, multipath routing offers additional routing options which can be employed to circumvent congested nodes in a timely manner, thereby reducing the delay. However, in large-scale constellations, there are always sufficient alternative routes, and the improvement in end-to-end delay metrics is not substantial.

The experimental results demonstrate that the SPF scheme with the shortest path exhibited a lower average delay in low-load scenarios. However, SPF exhibited the lowest average throughput and average flow completion ratio, and the performance of SPF deteriorated rapidly under high-load conditions. In contrast, the proposed DMR scheme exhibited a better average throughput and flow completion ratio under different loads, and it exhibited a better generalization ability over constellations of different sizes and topologies without repeated training.

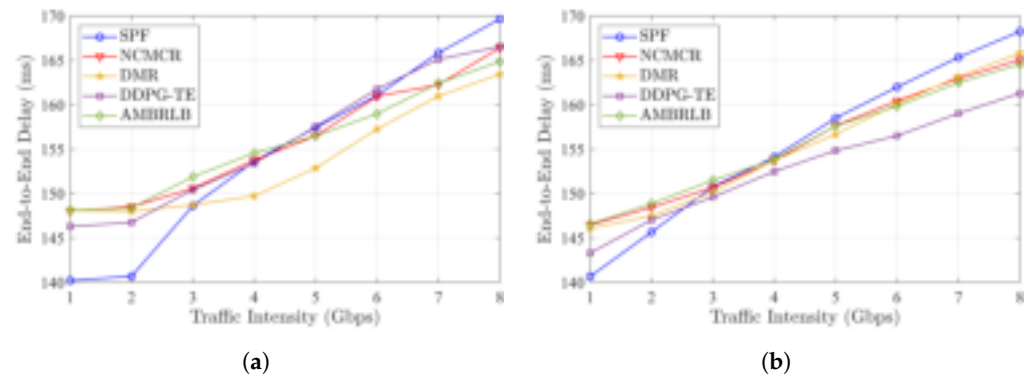


Figure 8. Average end-to-end delay for different constellations with various traffic intensities. (a) Average end-to-end delay for Iridium constellation. (b) Average end-to-end delay for OneWeb constellation.

6. Conclusions

This paper proposed a deep reinforcement learning (DRL)-based multipath routing (DMR) solution for satellite networks. The objective was to address the multipath routing transmission problem in LEO megaconstellation networks. The proposed DMR approach split multipath routing into two subproblems: multipath route selection and multipath traffic scheduling. In the context of the multipath path selection subproblem, this paper proposed a centralized routing scheme, MHMRD, which generates a set of available paths for end-to-end nodes. This is achieved by collecting node state information based on the current end-to-end minimum hop count and link utilization. In the multipath traffic scheduling subproblem, GNN-based multipath traffic scheduling (GMTS) is proposed. Each routing node is controlled by a DQN agent, and the GNN is used to address the issue of topologically varying satellite network topologies. The optimal multipath traffic scheduling model was obtained through iterative training. The proposed DMR scheme was finally validated on two different sizes of megaconstellations in the NS3 simulation environment. The simulation results demonstrate that the proposed DMR scheme exhibited superior performance in terms of the average network throughput and average flow completion ratio compared with the baseline scheme. Furthermore, the DMR scheme effectively addressed satellite network routing under load balancing while maintaining minimal increases in delay costs. Furthermore, the DMR scheme devised in this paper exhibited favorable scalability and adaptability in the context of evolving constellation topologies. It is also capable of being extended to satellite networks with changing topologies without the necessity for additional training.

It should be noted that as the size of the constellation continues to expand, the computational requirements of DQN-based online routing also rise. In the future, we intend to explore the potential of adopting solutions such as multi-controller deployment to enhance the responsiveness of the network and further validate its efficacy in real-world scenarios.

Author Contributions: Conceptualization, C.H.; methodology, C.H.; software, R.Y.; validation, W.X.; formal analysis, C.H.; investigation, W.X.; resources, W.X.; data curation, C.H.; writing—original draft preparation, C.H.; writing—review and editing, C.H.; visualization, C.H.; supervision, W.X.; project administration, W.X. and R.Y.; funding acquisition, W.X. and R.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Laboratory Fund Project under grant number 614201003022207 and the Aerospace Discipline Education New Engineering Project under grant number 145AXL250004000X.

Data Availability Statement: The relevant data and source code of this paper can be accessed by contacting hanchi@hgd.edu.cn.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DRL	Deep reinforcement learning
DMR	DRL-based multipath routing
MHMRD	Minimum hop count-based multipath routing discovery algorithm
GNN	Graph neural network
GMTS	GNN-based multipath traffic scheduling
MCN	Megaconstellation network
SAGIN	Space-air-ground integrated network
ISLs	Inter-satellite links
SDN	Software-defined network
RAAN	Right ascension of ascending node
NCC	Network control center
LEO	Low Earth orbit
DQN	Deep Q network

References

- Jiang, X.; Huang, Y.; Li, J.; He, H.; Chen, S.; Yang, F.; Yang, J. Spatio-Temporal Routing, Redundant Coding and Multipath Scheduling for Deterministic Satellite Network Transmission. *IEEE Trans. Commun.* **2023**, *71*, 2860–2875. [\[CrossRef\]](#)
- Kodheli, O.; Lagunas, E.; Maturo, N.; Sharma, S.K.; Shankar, B.; Montoya, J.F.M.; Duncan, J.C.M.; Spano, D.; Chatzinotas, S.; Kisseleff, S.; et al. Satellite Communications in the New Space Era: A Survey and Future Challenges. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 70–109. [\[CrossRef\]](#)
- Rakhmanov, A.; Wiseman, Y. Compression of GNSS Data with the Aim of Speeding up Communication to Autonomous Vehicles. *Remote Sens.* **2023**, *15*, 2165. [\[CrossRef\]](#)
- Correia, S.D.; Perez, R.; Matos-Carvalho, J.; Leithardt, V.R.Q. μ JSON, a Lightweight Compression Scheme for Embedded GNSS Data Transmission on IoT Nodes. In Proceedings of the 2022 5th Conference on Cloud and Internet of Things (CIoT), Marrakech, Morocco, 28–30 March 2022; pp. 232–238.
- Xue, C.; Li, W.; Yu, L.; Shang, J.; Chen, X.; Lu, S. SERO: A Model-Driven Seamless Roaming Framework for Wireless Mesh Network With Multipath TCP. *IEEE Trans. Commun.* **2019**, *67*, 1284–1296. [\[CrossRef\]](#)
- Liu, X.; Ma, T.; Qin, X.; Zhou, H.; Zhao, L. A DRL Empowered Multipath Cooperative Routing for Ultra-Dense LEO Satellite Networks. In Proceedings of the GLOBECOM 2023, Kuala Lumpur, Malaysia, 4–8 December 2023; pp. 5961–5966.
- Li, R.; Zhang, J.; Zheng, S.; Wang, K.; Wang, P.; Zhang, X. LEO Mega-Constellations Routing Algorithm Based on Area Segmentation. In Proceedings of the 2023 IEEE Wireless Communications and Networking Conference (WCNC), Glasgow, UK, 26–29 March 2023; pp. 1–6.
- Li, Y.; Zhang, Q.; Yao, H.; Gao, R.; Xin, X.; Yu, F.R. Stigmergy and Hierarchical Learning for Routing Optimization in Multi-Domain Collaborative Satellite Networks. *IEEE J. Select. Areas Commun.* **2024**, *42*, 1188–1203. [\[CrossRef\]](#)
- Perkins, C.E.; Royer, E.M. Ad-Hoc on-Demand Distance Vector Routing. In Proceedings of the WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, USA, 25–26 February 1999; pp. 90–100.
- Jiang, F.; Zhang, Q.; Yang, Z.; Yuan, P. A Space-Time Graph Based Multipath Routing in Disruption-Tolerant Earth-Observing Satellite Networks. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 2592–2603. [\[CrossRef\]](#)
- Huang, Y.; Yang, D.; Feng, B.; Tian, A.; Dong, P.; Yu, S.; Zhang, H. A GNN-Enabled Multipath Routing Algorithm for Spatial-Temporal Varying LEO Satellite Networks. *IEEE Trans. Veh. Technol.* **2024**, *73*, 5454–5468. [\[CrossRef\]](#)
- Huang, Y.; Jiang, X.; Chen, S.; Yang, F.; Yang, J. Pheromone Incentivized Intelligent Multipath Traffic Scheduling Approach for LEO Satellite Networks. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 5889–5902. [\[CrossRef\]](#)
- Tian, A.; Feng, B.; Zhou, H.; Huang, Y.; Sood, K.; Yu, S.; Zhang, H. Efficient Federated DRL-Based Cooperative Caching for Mobile Edge Networks. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 246–260. [\[CrossRef\]](#)
- Huang, X.; Yuan, T.; Qiao, G.; Ren, Y. Deep Reinforcement Learning for Multimedia Traffic Control in Software Defined Networking. *IEEE Netw.* **2018**, *32*, 35–41. [\[CrossRef\]](#)
- Lei, K.; Guo, P.; Wang, Y.; Wu, X.; Zhao, W. Solve Routing Problems with a Residual Edge-Graph Attention Neural Network. *Neurocomputing* **2022**, *508*, 79–98. [\[CrossRef\]](#)
- Zheng, X.; Huang, W.; Li, H.; Li, G. Research on Generalized Intelligent Routing Technology Based on Graph Neural Network. *Electronics* **2022**, *11*, 2952. [\[CrossRef\]](#)
- Wei, H.; Zhao, Y.; Xu, K. G-Routing: Graph Neural Networks-Based Flexible Online Routing. *IEEE Netw.* **2023**, *37*, 90–96. [\[CrossRef\]](#)
- Rusek, K.; Suarez-Varela, J.; Almasan, P.; Barlet-Ros, P.; Cabellos-Aparicio, A. RouteNet: Leveraging Graph Neural Networks for Network Modeling and Optimization in SDN. *IEEE J. Select. Areas Commun.* **2020**, *38*, 2260–2270. [\[CrossRef\]](#)
- Bi, Y.; Han, G.; Xu, S.; Wang, X.; Lin, C.; Yu, Z.; Sun, P. Software Defined Space-Terrestrial Integrated Networks: Architecture, Challenges, and Solutions. *IEEE Netw.* **2019**, *33*, 22–28. [\[CrossRef\]](#)

20. Hu, M.; Xiao, M.; Hu, Y.; Cai, C.; Deng, T.; Peng, K. Software Defined Multicast Using Segment Routing in LEO Satellite Networks. *IEEE Trans. Mob. Comput.* **2024**, *23*, 835–849. [\[CrossRef\]](#)
21. Han, Z.; Xu, C.; Zhao, G.; Wang, S.; Cheng, K.; Yu, S. Time-Varying Topology Model for Dynamic Routing in LEO Satellite Constellation Networks. *IEEE Trans. Veh. Technol.* **2023**, *72*, 3440–3454. [\[CrossRef\]](#)
22. Giambene, G.; Luong, D.K.; De Cola, T.; Le, V.A.; Muhammad, M. Analysis of a Packet-Level Block Coding Approach for Terrestrial-Satellite Mobile Systems. *IEEE Trans. Veh. Technol.* **2019**, *68*, 8117–8132. [\[CrossRef\]](#)
23. Chen, Z.; Zhou, W.; Wu, S.; Cheng, L. An Adaptive On-Demand Multipath Routing Protocol With QoS Support for High-Speed MANET. *IEEE Access* **2020**, *8*, 44760–44773. [\[CrossRef\]](#)
24. Tang, F.; Zhang, H.; Yang, L.T. Multipath Cooperative Routing with Efficient Acknowledgement for LEO Satellite Networks. *IEEE Trans. Mob. Comput.* **2019**, *18*, 179–192. [\[CrossRef\]](#)
25. Li, G.; Zhou, H.; Feng, B.; Zhang, Y.; Yu, S. Efficient Provision of Service Function Chains in Overlay Networks Using Reinforcement Learning. *IEEE Trans. Cloud Comput.* **2022**, *10*, 383–395. [\[CrossRef\]](#)
26. Kato, N.; Fadlullah, Z.M.; Mao, B.; Tang, F.; Akashi, O.; Inoue, T.; Mizutani, K. The Deep Learning Vision for Heterogeneous Network Traffic Control: Proposal, Challenges, and Future Perspective. *IEEE Wirel. Commun.* **2017**, *24*, 146–153. [\[CrossRef\]](#)
27. Mao, B.; Fadlullah, Z.M.; Tang, F.; Kato, N.; Akashi, O.; Inoue, T.; Mizutani, K. Routing or Computing? The Paradigm Shift Towards Intelligent Computer Network Packet Transmission Based on Deep Learning. *IEEE Trans. Comput.* **2017**, *66*, 1946–1960. [\[CrossRef\]](#)
28. Geyer, F.; Carle, G. Learning and Generating Distributed Routing Protocols Using Graph-Based Deep Learning. In Proceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks, Budapest, Hungary, 20 August 2018; pp. 40–45.
29. Liu, C.; Xu, M.; Yang, Y.; Geng, N. DRL-OR: Deep Reinforcement Learning-Based Online Routing for Multi-Type Service Requirements. In Proceedings of the IEEE INFOCOM 2021—IEEE Conference on Computer Communications, Vancouver, BC, Canada, 10–13 May 2021; pp. 1–10.
30. Almasan, P.; Suárez-Varela, J.; Rusek, K.; Barlet-Ros, P.; Cabellos-Aparicio, A. Deep Reinforcement Learning Meets Graph Neural Networks: Exploring a Routing Optimization Use Case. *Comput. Commun.* **2022**, *196*, 184–194. [\[CrossRef\]](#)
31. Sun, P.; Guo, Z.; Li, J.; Xu, Y.; Lan, J.; Hu, Y. Enabling Scalable Routing in Software-Defined Networks With Deep Reinforcement Learning on Critical Nodes. *IEEE/ACM Trans. Netw.* **2022**, *30*, 629–640. [\[CrossRef\]](#)
32. Kim, G.; Kim, Y.; Lim, H. Deep Reinforcement Learning-Based Routing on Software-Defined Networks. *IEEE Access* **2022**, *10*, 18121–18133. [\[CrossRef\]](#)
33. Jiang, Z.; Wu, Q.; Li, H.; Wu, J. scMPTCP: SDN Cooperated Multipath Transfer for Satellite Network With Load Awareness. *IEEE Access* **2018**, *6*, 19823–19832. [\[CrossRef\]](#)
34. De Santis, E.; Giuseppi, A.; Pietrabissa, A.; Capponi, M.; Delli Priscoli, F. Satellite Integration into 5G: Deep Reinforcement Learning for Network Selection. *Mach. Intell. Res.* **2022**, *19*, 127–137. [\[CrossRef\]](#)
35. Iridium Constellation. Available online: <https://celestrak.org/NORAD/elements/table.php?GROUP=iridium> (accessed on 30 July 2024).
36. Oneweb Constellation. Available online: <https://celestrak.org/NORAD/elements/table.php?GROUP=oneweb> (accessed on 30 July 2024).
37. MAWI Working Group Traffic Archive. Available online: <https://mawi.wide.ad.jp/mawi/> (accessed on 30 July 2024).
38. Francis Antony Selvi, P.; Manikandan, M.S.K. Ant Based Multipath Backbone Routing for Load Balancing in MANET. *IET Commun.* **2017**, *11*, 136–141. [\[CrossRef\]](#)
39. Chen, B.; Sun, P.; Zhang, P.; Lan, J.; Bu, Y.; Shen, J. Traffic Engineering Based on Deep Reinforcement Learning in Hybrid IP/SR Network. *China Commun.* **2021**, *18*, 204–213. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Disturbance Compensator Design Based on Dilated LMI for Linear Parameter-Varying Systems

Mustafa İnci ¹ and Yusuf Altun ^{2,*} ¹ Volta Motor Industry and Trade Inc. Design Center, Cumayeri Factory, 81700 Düzce, Türkiye; mustafa@volta.com.tr² Department of Computer Engineering, Düzce University, 81620 Düzce, Türkiye

* Correspondence: yusufaltun@duzce.edu.tr

Abstract: This paper presents a new dilated linear matrix inequality (LMI) representation to design a state feedback controller and a dynamic feedforward disturbance compensator for linear parameter-varying (LPV) systems. The improved LMIs are convex and finite-dimensional without any iterative approach. The designs are based on a new proposed equivalent bounded real lemma (BRL) by means of matrix dilation for LPV systems and uncertain linear systems under time-varying parametric uncertainties (TVPU). This dilated BRL provides lower conservative results than existing methods in terms of robust stability. Accordingly, a dynamic disturbance compensator is designed in addition to a state feedback controller. This paper mainly focuses on the design of compensators against disturbances in addition to the design of state feedback controllers. The dynamic matrices of the compensator change with the time-varying parameters of the LPV or uncertain system during operation, assuming that the disturbances and the parameters are measurable or observable. The compensator can be designed to attenuate the disturbances/noises or to improve reference tracking. Finally, numerical and simulation outcomes are presented to prove both the effectiveness and lower conservativeness of the proposed LMIs.

Keywords: LPV feedforward; disturbance compensator; H_∞ control; dilated BRL; robust control

**Citation:** İnci, M.; Altun, Y.

Disturbance Compensator Design

Based on Dilated LMI for Linear

Parameter-Varying Systems.

Electronics **2024**, *13*, 3055.<https://doi.org/10.3390/electronics13153055>

electronics13153055

Academic Editor: Luca Patané

Received: 5 July 2024

Revised: 26 July 2024

Accepted: 29 July 2024

Published: 1 August 2024

**Copyright:** © 2024 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the Creative Commons

Attribution (CC BY) license ([https://creativecommons.org/licenses/by/](https://creativecommons.org/licenses/by/4.0/)[https://creativecommons.org/licenses/by/](https://creativecommons.org/licenses/by/4.0/)

4.0/).

1. Introduction

Robust performance analyses of LPV systems, which are a special class of nonlinear system, and linear systems with time-varying parametric uncertainties (TVPU) have drawn considerable attention. Over the past two decades, many LMIs based on Lyapunov's stability criteria have been proposed for these systems since the efficiency of LMI-based methods is well known, such as [1–4]. Most of them are based on the standard quadratic Lyapunov function known as quadratic stability where a single quadratic Lyapunov function is used as in [5]. For example, a numerical comparison is made in terms of the conservativeness of quadratic stability conditions in [6]. However, many studies have shown that a parameter-dependent Lyapunov matrix (PDLM) overcomes the conservativeness derived from a single quadratic Lyapunov function for the systems such as those in [6–13]. However, it is not appropriate for LPV systems or the continuous-time system (CTS) under TVPU without an extra method since the PDLM or parameterized LMI (PLMI) causes a nonconvex optimization problem. However, the PDLM decreases the conservatism arising from the quadratic stability conditions. To address this problem, some approaches have been proposed such as the grid method [14], the s-procedure [15], the multi-convexity method [15,16], and the methods in [17,18]. Nevertheless, the grid method is valid only for local operation conditions contrary to all-admissible parameter trajectories. In addition, the other approaches involve the extra LMI conditions, some restrictions or complexity apart from the existing LMIs to obtain a convex solution when the PDLM is used. Furthermore, these approaches do not enable the design of multi-purpose control

in [12]. Moreover, a state feedback controller designed by using the methods depends on a Lyapunov function that is restricted by the symmetric property.

Recently, to provide lower conservativeness for uncertain and LPV systems, Lyapunov matrices depending on the parameters or PLMIs have been researched without extra LMI conditions. Accordingly, “dilated” or “extended” LMIs have been suggested to address nonconvex problems and have been very popular over the past two decades. S-variables and slack-variable LMIs (SV-LMIs) are also known since slack variables are generally used for LMIs [19]. The benefit of dilated LMIs is that lower conservative results are obtained for very large parameter sets since they do not contain the multiplication of the Lyapunov matrix with the state-space matrices of the system in contrast to the common LMIs as in [5]. This case also allows us to check the robust performance or stability, and a convex LMI formulation is obtained for the systems owing to the separation. In addition, dilated LMIs have been valuable tools for designing multi-objective controllers with lower conservativeness, as in [20]. In brief, the dilated LMI approach presents a convex solution and lower conservative results for the LPV or uncertain linear systems without any extra conditions despite the PDLMI, which leads to nonconvexity or infinite LMIs in common designs. For example, a stability condition is proposed in [8,21], but an H_∞ control design is not addressed. In [20], an obvious benefit of dilated LMIs over classical LMIs is presented for designing multi-objective H_2/D -stability controllers, but this formulation cannot be extended to H_∞ control synthesis. In [22], H_2 control via a slack variable is proposed, but H_∞ control is not addressed. In [23], lower conservative dilated LMI conditions are proposed under rationally varying parameters, but they are not appropriate for a CTS with TVPU and LPV systems. Apart from these studies, the improvements for the state feedback control in [24–31] by using dilated LMIs are more remarkable in the literature in terms of conservativeness and/or simplicity. These studies are less conservative than the standard state feedback controller design in [1]. However, none of them is fully appropriate for LPV systems and CTS under the TVPU whereas they are improved for CTS under time-invariant parametric uncertainties (TIPUs). Additionally, the proposed dilated LMI outperforms TIPU in terms of conservativeness. Accordingly, this specific topic is still open in the literature especially for the CTS with TVPU and LPV systems. Therefore, the first contribution of this paper is that a new dilated BRL is proposed for LPV systems and CTS with TVPU. This is subsequently extended to solve the state feedback H_∞ control problem. Therefore, less conservative results are obtained for LPV systems or the CTS with TVPU according to existing results in the literature.

Feedforward compensators or controllers are used to track the reference signal or to attenuate the external or internal disturbances, which are mostly predictable or measurable as in [32,33]. The performance of the system is improved owing to an extra generated control signal in accordance with the measured disturbance or reference values. In particular, disturbance compensators have been realized for many systems, such as [34]. A disturbance compensator is used together with a feedback controller as in [35,36], because the feedback design guarantees plant stability. However, the compensator does not provide stability to the unstable system. There are two main methods. The first is that they are concurrently designed as in [37,38]. For the second one, a certain performance is previously achieved with the feedback controller, and then the feedforward part is designed as in [39]. This paper is based on the second approach. Many feedforward compensators or controllers have been presented for LTI systems such as those in the above studies. There are a few works via the H_∞ method, especially for LPV systems, as in [40–45]. In [40], an H_∞ controller, which is reduced-order, has been designed to address the disturbances of the vibration system. In [41], a dynamic H_∞ feedforward controller design has been addressed by obtaining the system inverse. In [42], switched robust feedforward control has been presented for a class of uncertain linear systems defined in a classic linear fractional transformation form. However, none of them is appropriate for LPV systems. In [43,44], static LPV feedforward designs are used, but they are not dynamic controllers. A dynamic feedforward controller has been designed for LPV systems in [45], but many LMIs exist because some extra LMI

conditions are needed because of nonconvexity from using the PDLM. In addition, the LMIs are based on a common BRL, so LMIs are somewhat conservative because of the multiplication of the Lyapunov matrix. Therefore, the feedforward controller or compensator based on L_2 gain has been rarely discussed, especially for LPV systems. Moreover, a dynamic feedforward controller or compensator design has not been used in the above studies, achieving the advantages of the dilated LMI. Therefore, the second contribution of this paper is the use of a dynamic LPV feed-forward compensator based on an L_2 -gain controller design to achieve the advantages of the extended LMI approach, which according to the literature survey has not been reported in the literature so far. In [45], which is closer to this paper, the study is quite complex because of the occurrence of infinite LMIs. In addition, the design is based on standard BRL, which produces more conservative results.

In brief, in this study, solutions based on noniterative LMIs are proposed to design an LPV state feedback controller and a dynamic LPV feedforward compensator. They are convenient for both LPV and CTS with TVPU (linear systems with parametric uncertainty). The main contributions of the paper are as follows:

- (1) A dilated BRL based on a noniterative LMI is proposed for the CTS with TVPUs and LPV system by providing the advantages of a slack variable and dilated LMI. Then, a new state feedback controller is designed on the basis of a new equivalent BRL. With respect to its superiority, it has the best result among recent studies in [5,24–30] in view of its being less conservative as shown in Table 1, and an approximately 10% improvement has been achieved from the best result among them.
- (2) All proposed solutions do not need any additional LMI conditions as in [45] owing to the proposed dilated BRL even if the PDLM is used, which leads to nonconvexity or infinite LMIs. They are simpler and less conservative than the common approaches in the literature and less conservative than similar studies in the literature as proven via numerical tests. Therefore, the LMIs provide both simplicity and lower conservativeness.
- (3) A dynamic LPV feedforward compensator based on the dilated BRL is proposed for LPV systems with measurable or observable disturbance or noise. Therefore, the compensator works online with the time-varying parametric uncertainties of uncertain systems or the time-varying parameters for LPV systems.

Table 1. Comparisons of the achieved $\gamma_{\min}$.

	[5]	[24]	[25]	[26]	[27]	[28]	[29]	[30]	Proposed Theorem 1
$\gamma_{\min}$	1.557	1.498	1.478	1.476	1.251	1.241	1.240	1.241	1.118

2. Preliminaries

In this section, some definitions and lemmas are given to shed light on the proposed lemma and theorems.

Definition 1. The system described in (1) is named an LPV system. Here, $x(t) \in \mathbb{R}^{n_w}$ represents the states, $u(t) \in \mathbb{R}^{n_u}$ represents the control input signals, $z(t) \in \mathbb{R}^{n_z}$ represents the controlled outputs, $y(t) \in \mathbb{R}^{n_y}$ represents the measured outputs, $\omega(t) \in \mathbb{R}^{n_w}$ represents the disturbances and the whole matrices are of appropriate size and dependent on α parameters changing with time. The parameter sets are defined by (2) where Δ_R and Δ_D denote the parameter sets for α and its derivative, respectively. Additionally, the matrix $A(\alpha)$ is the state matrix that defines the relationship between the current state and its rate of change, $B_1(\alpha)$ is the disturbance input matrix, $B_2(\alpha)$ is the control input matrix, $C_1(\alpha)$ is the matrix that relates the state vector $x(t)$ to the controlled outputs $z(t)$, $D_{11}(\alpha)$ is the feedforward matrix for the disturbance input $\omega(t)$ directly linking the disturbance input $\omega(t)$ to the controlled output $z(t)$, $D_{12}(\alpha)$ is the feedforward matrix for the control input $u(t)$ directly linking the control input to the controlled output $z(t)$, $C_2(\alpha)$ is the measured output matrix defining the relationship between the state and the measured out-

put, $D_{21}(\alpha)$ is the feedforward matrix for the disturbance input directly linking the disturbance input $\omega(t)$ to the measured output $y(t)$ and $D_{22}(\alpha)$ is the feedforward matrix for the control input $u(t)$ directly linking the control input to the measured output $y(t)$.

$$\begin{aligned}\dot{x}(t) &= A(\alpha(t))x(t) + B_1(\alpha(t))\omega(t) + B_2(\alpha(t))u(t) \\ z(t) &= C_1(\alpha(t))x(t) + D_{11}(\alpha(t))\omega(t) + D_{12}(\alpha(t))u(t) \\ y(t) &= C_2(\alpha(t))x(t) + D_{21}(\alpha(t))\omega(t) + D_{22}(\alpha(t))u(t)\end{aligned}\quad (1)$$

$$\begin{aligned}\Delta_R &= \{\alpha \in \mathbb{R}^n : \bar{\alpha}_i \leq \alpha_i \leq \underline{\alpha}_i, \forall i = 1, \dots, n\} \\ \Delta_D &= \{\dot{\alpha} \in \mathbb{R}^n : \bar{\dot{\alpha}}_i \leq \dot{\alpha}_i \leq \underline{\dot{\alpha}}_i, \forall i = 1, \dots, n\}\end{aligned}\quad (2)$$

In addition, the closed-loop form of the system can be written as in (3) where Z indicates the outputs z and y . Additionally, the matrices A , B , C and D are considered as the closed-loop matrices.

$$\begin{aligned}\dot{x}(t) &= A(\alpha(t))x(t) + B(\alpha(t))\omega(t) \\ Z(t) &= C(\alpha(t))x(t) + D(\alpha(t))\omega(t)\end{aligned}\quad (3)$$

It is assumed that the parameters are available online (measurable or computable). Additionally, the parameters are derived from the states of the nonlinear system although it normally looks like an uncertain system. If the parameters are not derived from states, unlike those of the LPV system, (1) is the linear system under time-varying uncertainties as in Example 1.

The H_∞ norm for the system in (3) can be calculated via the common LMI formulation given by Lemma 1, which is well known as the BRL in the literature. Accordingly, for a positive scalar γ , we define the performance index by (4), where $\omega(t) \in L_2^q(0, \infty)$ is an exogenous disturbance.

$$J(\omega, z) = \left(\int z^T z - \gamma^2 \omega^T \omega \right) d\tau \quad (4)$$

Lemma 1. Let $G(\alpha)$ denote the stable transfer matrix of the LPV system in (3) for all admissible parameter trajectories. If and only if $\gamma^2 I - D^T(\alpha)D(\alpha) \succ 0$, $G(s)_\infty \prec \gamma$ and the following statements are equivalent for all $\forall(\alpha) \in \Delta_R$ and $\forall(\dot{\alpha}) \in \Delta_D$.

(i) If there exists a positive definite differentiable symmetric matrix $P(\alpha) \in \mathbb{R}^{n \times n}$, then

$$\begin{aligned}A^T(\alpha)P(\alpha) + P(\alpha)A(\alpha) + \dot{P}(\alpha) + C^T(\alpha)C(\alpha) \\ + (B^T(\alpha)P(\alpha) + D^T(\alpha)C(\alpha))^T (\gamma^2 I - D^T(\alpha)D(\alpha))^{-1} (B^T(\alpha)P(\alpha) + D^T(\alpha)C(\alpha)) \prec 0\end{aligned}\quad (5)$$

(ii) If there exists a positive definite differentiable symmetric matrix $P(\alpha) \in \mathbb{R}^{n \times n}$, then

$$\begin{pmatrix} A^T(\alpha)P(\alpha) + P(\alpha)A(\alpha) + \dot{P}(\alpha) & P(\alpha)B(\alpha) & C^T(\alpha) \\ B^T(\alpha)P(\alpha) & -\gamma^2 I & D^T(\alpha) \\ C(\alpha) & D(\alpha) & -I \end{pmatrix} \prec 0 \quad (6)$$

Remark 1. LMIs such as those in (6) are not convenient to design controllers for linear systems with polytopic or affine-type uncertain parameters or LPV systems since the inequality involves the multiplication of the Lyapunov matrix by the matrices of the plant. Therefore, this affair results in a nonconvex optimization and the infinite LMIs. Nevertheless, some methods have been suggested to eliminate this difficulty such as the grid method [14], the s -procedure [15], the multi-convexity method [11] and methods in [17,18]. However, the multi-convexity method and the s -procedure result in many LMIs, and the grid method is only valid for local operation conditions in contrast to all-admissible parameter trajectories. In addition, such techniques do not enable the design of a multi-purpose controller in [12]. The use of the parameter-independent (fixed) Lyapunov function

in (7) as in [5,41,46] is common, but this causes highly conservative results. Conversely, when the Lyapunov function is isolated from the plant matrices by using a slack variable, which is generally called the dilated or extended LMI approach due to the enlarging dimension of the LMI, this enables one to diminish the conservatism for analysis and design, as in [23,24,27,31]. Therefore, attaining a dilated condition equivalent to the BRL is an effective approach to overcome this complexity in robust control designs such as H_2 and H_∞ . Accordingly, a new notation for H_∞ computation including the TVPU has been proposed in Lemma 3 for controller design.

The standard LMI in (7) derived from Lemma 1 is generally employed for the L_2 -gain or H_∞ design ($K(\alpha) = R(\alpha)P^{-1}$) for LPV systems in the literature [5].

$$\begin{pmatrix} A^T(\alpha)P + B_2R(\alpha) + (*)^T & B_1(\alpha) & PC_1^T(\alpha) + R^T(\alpha)D_{12}^T \\ * & -\gamma^2 I & D_{11}^T(\alpha) \\ * & * & -I \end{pmatrix} \prec 0 \quad (7)$$

The following Lemma 2, which is based on the projection lemma, is useful for deriving the dilated LMIs.

Lemma 2. [21]: If there exist a symmetric matrix Ω and compatible dimensional matrices Θ_1, Θ_2 , then, the following conditions are equivalent:

- i. $\Omega < 0$ and $\Omega + \Theta_1\Theta_2^T + \Theta_2\Theta_1^T < 0$
- ii. The following LMI problem is feasible in accordance with Q .

$$\begin{pmatrix} \Omega & \Theta_2 + \Theta_1 Q \\ \Theta_2^T + Q^T \Theta_1^T & -Q - Q^T \end{pmatrix} \prec 0$$

3. A New Dilated Bounded Real Lemma

In this section, a dilated BRL is proposed for the LPV and uncertain systems. Accordingly, an LMI-based solution is improved to design the H_∞ state feedback controller. Lemma 3 shows the dilated BRL.

Lemma 3. Let $G(\alpha)$ denote the transfer matrix of the LPV system in (3), then the following two statements are equivalent.

- i. If and only if the matrix $A(\alpha)$ is stable for all admissible parameter trajectories and $D^T(\alpha)D(\alpha) - \gamma^2 I \prec 0$, the H_∞ performance for the system in (3) is bounded such that the index $\gamma > 0$. Thus, $\|G_{z\omega}\|_\infty \prec \gamma$.
- ii. If there exist a positive definite parameter-dependent differentiable symmetric matrix $P(\alpha) \in \mathbb{R}^{n \times n}$, a real positive constant μ satisfying $\dot{P}(\alpha) - 2\mu P(\alpha) \prec 0$ and a compatible dimensional matrix Q , then the LMI problem in (8) holds for all $\forall \alpha \in \Delta_R$ and $\forall \dot{\alpha} \in \Delta_D$.

$$\begin{pmatrix} \dot{P}(\alpha) - 2\mu P(\alpha) & 0 & P(\alpha) + A^T(\alpha)Q + \mu Q & C^T(\alpha) \\ * & -\gamma^2 I & B^T(\alpha)Q & D^T(\alpha) \\ * & * & -Q - Q^T & 0 \\ * & * & * & -I \end{pmatrix} \prec 0 \quad (8)$$

Proof of Lemma 3. The LPV form of the inequality in statement i of Lemma 1 is as follows.

$$\begin{aligned} & A^T(\alpha)P(\alpha) + P(\alpha)A(\alpha) + \dot{P}(\alpha) + C^T(\alpha)C(\alpha) \\ & + (B^T(\alpha)P(\alpha) + D^T(\alpha)C(\alpha))^T (\gamma^2 I - D^T(\alpha)D(\alpha))^{-1} (B^T(\alpha)P(\alpha) + D^T(\alpha)C(\alpha)) \prec 0 \end{aligned} \quad (9)$$

The inequality in (9) is modified as follows, when a real constant variable μ is included in the inequality.

$$\begin{aligned} & (A^T(\alpha) + \mu I)P(\alpha) + P(\alpha)(A(\alpha) + \mu I) + C^T(\alpha)C(\alpha) + \dot{P}(\alpha) - 2\mu P(\alpha) \\ & + (B^T(\alpha)P(\alpha) + D^T(\alpha)C(\alpha))^T (\gamma^2 I - D^T(\alpha)D(\alpha))^{-1} (B^T(\alpha)P(\alpha) + D^T(\alpha)C(\alpha)) \prec 0 \end{aligned}$$

By using the Schur lemma, a new matrix inequality is obtained as follows.

$$\begin{pmatrix} \Gamma_{11}(\alpha) & \Gamma_{12}(\alpha) \\ \Gamma_{13}(\alpha) & \Gamma_{14}(\alpha) \end{pmatrix} \prec 0$$

where

$$\begin{aligned} \Gamma_{11}(\alpha) &= (A^T(\alpha) + \mu I)P(\alpha) + P(\alpha)(A(\alpha) + \mu I) + C^T(\alpha)C(\alpha) + \dot{P}(\alpha) - 2\mu P(\alpha) \\ \Gamma_{12}(\alpha) &= P(\alpha)B(\alpha) + C^T(\alpha)D(\alpha), \\ \Gamma_{13}(\alpha) &= B^T(\alpha)P(\alpha) + D^T(\alpha)C(\alpha) \\ \Gamma_{14}(\alpha) &= D^T(\alpha)D(\alpha) - \gamma^2 I \end{aligned}$$

We can rewrite the inequality as follows.

$$\begin{pmatrix} (A^T(\alpha) + \mu I)P(\alpha) + (*)^T & P(\alpha)B(\alpha) \\ B^T(\alpha)P(\alpha) & 0 \end{pmatrix} + \begin{pmatrix} C^T(\alpha)C(\alpha) + \dot{P}(\alpha) - 2\mu P(\alpha) & C^T(\alpha)D(\alpha) \\ D^T(\alpha)C(\alpha) & D^T(\alpha)D(\alpha) - \gamma^2 I \end{pmatrix} \prec 0 \quad (10)$$

This is rewritten as follows.

$$\begin{pmatrix} C^T(\alpha)C(\alpha) + \dot{P}(\alpha) - 2\mu P(\alpha) & C(\alpha)^T D(\alpha) \\ D^T(\alpha)C(\alpha) & D^T(\alpha)D(\alpha) - \gamma^2 I \end{pmatrix} + \begin{pmatrix} A^T(\alpha) + \mu I \\ B^T(\alpha) \end{pmatrix} \begin{pmatrix} P(\alpha) & 0 \end{pmatrix} + \begin{pmatrix} P(\alpha) \\ 0 \end{pmatrix} \begin{pmatrix} A(\alpha) & B(\alpha) \\ +\mu I & \end{pmatrix} \prec 0 \quad (11)$$

Afterwards, we can obtain the inequality (12) if we apply Lemma 2 to (11).

$$\begin{pmatrix} \varphi_{11}(\alpha) & \varphi_{12}(\alpha) & \varphi_{13}(\alpha) \\ * & \varphi_{22}(\alpha) & \varphi_{23}(\alpha) \\ * & * & \varphi_{33} \end{pmatrix} \prec 0 \quad (12)$$

where

$$\begin{aligned} \varphi_{11}(\alpha) &= C^T(\alpha)C(\alpha) + \dot{P}(\alpha) - 2\mu P(\alpha), \\ \varphi_{12}(\alpha) &= C^T(\alpha)D(\alpha) \\ \varphi_{13}(\alpha) &= P(\alpha) + (A^T(\alpha) + \mu I)Q, \\ \varphi_{22}(\alpha) &= D^T(\alpha)D(\alpha) - \gamma^2 I \\ \varphi_{23}(\alpha) &= B^T(\alpha)Q, \quad \varphi_{33} = -Q - Q^T \end{aligned}$$

However, the expression on the left side of (11) must be negative definite according to Lemma 2. Accordingly, we rewrite the expression including $\dot{P}(\alpha) - 2\mu P(\alpha) \prec 0$ as follows.

$$\begin{pmatrix} \dot{P}(\alpha) - 2\mu P(\alpha) & 0 \\ * & -\gamma^2 I \end{pmatrix} + \begin{pmatrix} C^T(\alpha) \\ D^T(\alpha) \end{pmatrix} I(C(\alpha) \quad D(\alpha)) \prec 0$$

Therefore, we obtain the inequality (13) via the Schur lemma.

$$\begin{pmatrix} \dot{P}(\alpha) - 2\mu P(\alpha) & 0 & C^T(\alpha) \\ * & -\gamma^2 I & D^T(\alpha) \\ * & * & -I \end{pmatrix} \prec 0 \quad (13)$$

Finally, we obtain (8) if we apply the Schur lemma after rewriting the inequality (12) as the following form. In addition, the condition (8) also covers the condition (13).

$$\begin{pmatrix} \dot{P}(\alpha) - 2\mu P(\alpha) & 0 & P(\alpha) + A^T(\alpha)Q + \mu Q \\ * & -\gamma^2 I & B^T(\alpha)Q \\ * & * & -Q - Q^T \end{pmatrix} + \begin{pmatrix} C^T(\alpha) \\ D^T(\alpha) \\ 0 \end{pmatrix} I \begin{pmatrix} C(\alpha) & D(\alpha) & 0 \end{pmatrix} \prec 0$$

Hence, the proof is complete. $\square$

All the designs can be used for polytopic-type or affine-type parameters by making some regulations. However, affine-type varying parameters are used for the examples throughout this paper, and thus, all the time-varying matrices affinely change such as the Lyapunov matrix in (14). Therefore, affine quadratic stability is obtained throughout the paper as in [11].

$$P(\alpha) = P_0 + \sum_{i=1}^n \alpha_i P_i, \quad \dot{P}(\alpha) = \sum_{i=1}^n \dot{\alpha}_i P_i \quad (14)$$

Theorem 1. Let the LPV system be in (1) with the time-varying parameter set in (2). If there exist a positive real scalar μ satisfying $\dot{L}(\alpha) - 2\mu L(\alpha) \prec 0$, the symmetric positive definite differentiable matrix $L(\alpha)$ with the compatible dimension matrices R , $F(\alpha)$, then the LMI in (15) holds for $\forall(\alpha) \in \Delta_R$ and $\forall(\alpha) \in \Delta_D$. Thus, there exists a state feedback H_∞ controller $K(\alpha) = F(\alpha)R^{-1}$.

$$\begin{pmatrix} \dot{L}(\alpha) - 2\mu L(\alpha) & * & * & * \\ 0 & -\gamma^2 I & * & * \\ L(\alpha) + A(\alpha)R + B_2F(\alpha) + \mu R & B_1(\alpha) & -R - R^T & * \\ C_1(\alpha)R + D_{12}F(\alpha) & D_{11}(\alpha) & 0 & -I \end{pmatrix} \prec 0 \quad (15)$$

Proof of Theorem 1. We obtained the closed-loop matrices in (16) as in (17) when the controller $u = K(\alpha)x$ is substituted in the system (1) where the output $y(t)$ is eliminated since it is considered state feedback.

$$\begin{aligned} \dot{x}_{cl}(t) &= A_{cl}(\alpha(t))x_{cl}(t) + B_{cl}(\alpha(t))\omega(t), \\ z_{cl}(t) &= C_{cl}(\alpha(t))x_{cl}(t) + D_{cl}(\alpha(t))\omega(t) \end{aligned} \quad (16)$$

$$\begin{aligned} A_{cl}(\alpha) &= A(\alpha) + B_2K(\alpha), \quad B_{cl}(\alpha) = B_1(\alpha), \\ C_{cl}(\alpha) &= C(\alpha) + D_{12}K(\alpha), \quad D_{cl}(\alpha) = D_{11}(\alpha) \end{aligned} \quad (17)$$

If the state space matrices in (17) are replaced in (8), we obtain the following inequality.

$$\begin{pmatrix} \dot{P}(\alpha) - 2\mu P(\alpha) & * & * & * \\ 0 & -\gamma^2 I & * & * \\ P(\alpha) + Q^T(A(\alpha) + B_2K(\alpha)) + \mu Q^T & Q^TB_1(\alpha) & -Q - Q^T & * \\ C(\alpha) + D_{12}K(\alpha) & D_{11}(\alpha) & 0 & -I \end{pmatrix} \prec 0$$

However, there are multiplications of unknown matrices. To obtain the LMI, we apply congruence transformation to the inequality by multiplying the transpose of (18) and (18), respectively. Thus, (15) is obtained, where $L(\alpha) = Q^{-T}P(\alpha)Q^{-1}$, $\dot{L}(\alpha) = Q^{-T}\dot{P}(\alpha)Q^{-1}$, $R = Q^{-1}$, $F(\alpha) = K(\alpha)R$. Hence, the proof is complete. $\square$

$$\text{diag}(Q^{-1}, I, Q^{-1}, I) \quad (18)$$

Remark 2. The proposed solution provides three significant advantages. First, contrary to (7), the Lyapunov matrix can be in the form of a parameter-dependent matrix without extra approaches such

as the *s*-procedure, grid and multi-convexity, which reduces conservativeness. Second, contrary to the common design as in (7) in [5], the controller gain is independent of the Lyapunov matrix, which is restricted by the symmetric property. Instead, it changes with a nonsymmetric matrix R for the proposed LMI in Theorem 1. Third, the real slack variable μ yields feasible outcomes even for large ranges of parameters because a large increase in the range affects the γ performance value less than similar values in terms of feasibility and performance, as proven in the numerical results.

If we consider CTS with TVPU, we can easily change the LMI for parameter-independent (static or fixed) controller design as in Theorem 1 assuming that the uncertain parameters are both unknown at the design time and not known by online measurement but known to be bounded. Thus, some matrices are converted to parameter-dependent matrices whereas some matrices are converted to parameter-independent matrices to obtain the form of the LMI. In addition, the theorem can be used for the affine-type or polytopic-type time-varying parameters or uncertainties by making some regulations. However, affine-type time-varying parameters are considered for the designs throughout this paper. Two numerical examples have been realized for the proposed state feedback controller. All the numerical computations were performed via the Yalmip parser [47] and Sedumi solver [48], which were used on a PC (2 cores 2.8 GHz CPU with 4GB RAM). Example 1 presents a controller design for the CTS with the TIPU, whereas Example 2 presents a gain-scheduled control for the CTS with TVPU.

Example 1. The state feedback controller for the satellite in [25,49] is addressed. The system arises from two rigid bodies joined by a flexible link. The system is as in (19). The uncertain parameters are modeled as damping f and spring k . The range of k is [0.09 0.4] whereas the range of f is [0.0038 0.04]. The performances of the methods in [5,24–30] and the proposed solution in Theorem 1 are compared in terms of level γ which is directly related to conservativeness. The derivatives of the parameters are zero; that is, the uncertainties are not differentiable.

$$\begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k & k & -f & f \\ k & -k & f & -f \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \omega + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} u, \quad (19)$$

$$z = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \omega + \begin{pmatrix} 0 \\ 0.01 \end{pmatrix} u$$

Table 1 shows the minimal γ values for the results in [5,24–30] and the proposed solution in Theorem 1 for this example. The results indicate that the proposed solution has much better results than those of the other methods. The feedback gain of the common solution is $K = -10^5[6.1116 \ 44.042 \ 1.1054 \ 81.096]$, whereas that of the proposed solution is $K = -[205.82 \ 2224.6 \ 58.393 \ 2850.9]$. The superiority of the proposed solution is that it is less conservative than [5,24–30], as in Table 1. In addition, Figure 1a shows the relationship between μ and γ . The γ level is almost never affected by changes in μ after a certain value for this system. Moreover, the proposed Theorem 1 has the best result even if μ is between 2.5 and 35, as shown in Figure 1a.

Remark 3. The proposed solution for H_∞ robust control can be used for the CTS with TVPU by obtaining advantages of dilated LMI/slack variables, but the other solutions in [5,24–30] can be only used for the CTS with TIPU. Additionally, the LMI is noniterative. Therefore, the noniterative dilated LMI solution includes both TIPU and TVPU by providing the advantages of the dilated LMI. Its existence has not been reported in the literature. Accordingly, Example 1 presents a controller design for the CTS with the TIPU whereas Example 2 presents a controller design for the CTS with

the TVPU. Finally, Example 2 presents only a comparison between this paper and the standard method in [5] since the studies in Table 1 except for the standard design in [1] are not fully valid for the TVPU. The performances are tested by enlarging the uncertain set for the TVPU in Example 2.

Example 2. In this example, the gain scheduling LPV control is designed by using Theorem 1 for the randomly generated LPV system in (20) where the ranges of the time-varying parameters α_1 and α_2 are $[0.01, 2]$ and $[0.5, 1.5]$, respectively. Additionally, the rates of variation for the parameters $\dot{\alpha}_1$ and $\dot{\alpha}_2$ are $[0, 3]$ and $[0, 1]$, respectively.

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} -\alpha_1 & \alpha_1 & 0.8 \\ 0 & 0.2 & -\alpha_2 - 1 \\ 0.5 & \alpha_2 & 0.1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.2 \\ 0 \end{pmatrix} \omega + \begin{pmatrix} 1 \\ 0 \\ 0.1 \end{pmatrix} u, \quad (20)$$

$$z = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \omega + \begin{pmatrix} 0 \\ 0.8 \end{pmatrix} u$$

As in Table 2, the improvement substantially increases as the uncertain set enlarges compared with the common set. For example, the method decreases the γ level at a rate of 78.59% for $\alpha_1 = [0.01 \ 50]$, $\alpha_2 = [0.1 \ 190]$.

Table 2. The $\gamma_{\min}$ performance values for TVPU.

γ Values									
$\alpha_1 = [0.01 \ 2], \alpha_2 = [0.5 \ 1.5]$				$\alpha_1 = [0.01 \ 5], \alpha_2 = [0.1 \ 19]$			$\alpha_1 = [0.01 \ 50], \alpha_2 = [0.1 \ 190]$		
[5]	1.0280			6.3151			22.745		
Theorem 1	0.890 ($\mu = 3$)			2.2466 ($\mu = 2.5$)			4.869 ($\mu = 3.5$)		
	14.42% decreased			64.42% decreased			78.59% decreased		
Slack variable sets for feasibility									
Theorem 1	$\mu = (0, 9 \times 10^5]$			$\mu = (0, 8 \times 10^5]$			$\mu = (0, 8 \times 10^5]$		
	Number of iterations—Number of variables—Solver time								
[5]	15	18	0.0174 s	28	18	0.126 s	49	18	0.215 s
Theorem 1	39	41	9.556 s	41	41	9.904 s	56	41	13.503 s

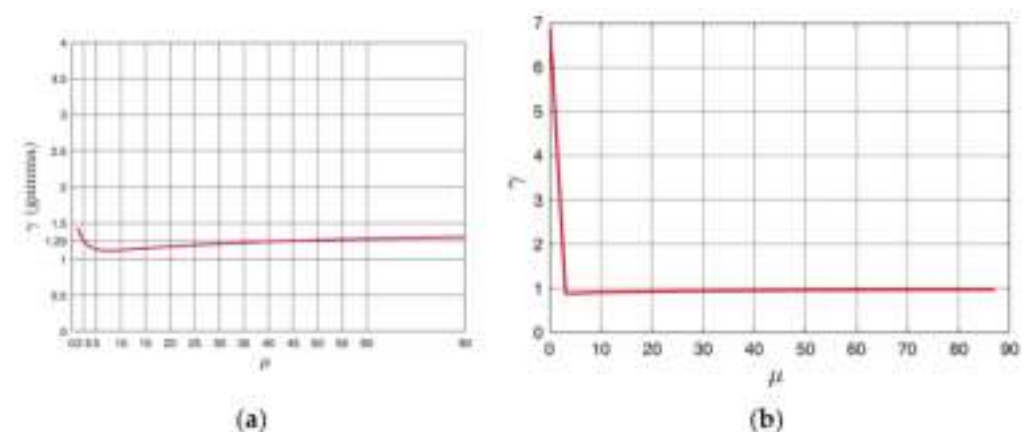


Figure 1. The relations between γ level and slack variable: (a) Example 1; (b) Example 2.

In addition, the relationship between μ and γ is shown in Figure 1b. Therefore, the γ level is almost never changed by changing μ after a certain value, so its determination is easy such that the proposed solution has better performance than the common solution even if μ is between 1 and 85.

With reference to this, the results are obtained for the proposed solution in Theorem 1 and the common design in [5], where the Lyapunov matrix is fixed. For $\alpha_1 = [0.01 \ 2]$, $\alpha_2 = [0.5 \ 1.5]$, the common design yields a γ level of 1.028, whereas the proposed method yields 0.890 for $\mu = 3$. Accordingly, the feedback gains of the standard design and the design in Theorem 1 are $K_0 = 10^3[-23.383 \ 170.568 \ -269.482]$, $K_1 = [0.30251 \ -2.2045 \ 3.4833]$, $K_2 = [0.11503 \ -0.84184 \ 1.33]$ and $K_0 = [-12.17 \ 74.915 \ -102.05]$, $K_1 = [0.9135 \ -3.0938 \ 3.5387]$, $K_2 = [5.306 \ -40.69 \ 47.744]$, respectively. The relationship between μ and γ is shown in Figure 1b.

From the results, the proposed dilated LMI solution is less conservative than the common LMI in [5] for the gain scheduling design although it has disadvantages which are a greater number of iterations, number of variables and solver time. The feasibility is obtained for the large uncertain set in Table 2, and the slack variable μ provides feasible results for a large set $(0, 8 \times 10^5]$. Although the number of LMI variables increased from 18 to 41 according to the standard one in [5], an improvement of up to 78.59% has been achieved. Finally, good improvement and less conservative results are achieved for the TVPU even if the dilated LMIs are more costly than the standard LMI is. Notably, the comparisons in the example are made for the parameter-independent Lyapunov matrix in [5]. As emphasized in Remark 1, if PLDF is used, methods such as the multi-convexity method [11] and methods in [17,18] are used. Therefore, the standard LMI variables and the complexity of [5] increase substantially. For example, another LMI condition of the same size needs to be added. Hence, the proposed LMI condition is considered to be less costly since it has a single LMI condition.

Two examples have shown that the proposed solution presents less conservative results. Furthermore, it can be fully used for the LPV system and the uncertain system under the TVPU, but the others cannot be used except for the common LMI in [5]. On the other hand, increasing bounds for time-varying uncertainties or parameters increase the conservativeness of LMIs. However, the design based on the proposed LMI achieves less conservative and more successful results in addition to a simpler structure.

4. Dynamic Disturbance Compensator Design for LPV Systems

The block diagram of the disturbance compensator design is shown in Figure 2 where NCF is the normalization coefficient or filter. Here, the dynamic feedforward compensator is as in (21).

$$\begin{aligned}\dot{x}_c(t) &= A_c(\alpha)x_c(t) + B_c(\alpha)\omega(t) \\ u_c(t) &= C_c(\alpha)x_c(t) + D_c(\alpha)\omega(t)\end{aligned}\quad (21)$$

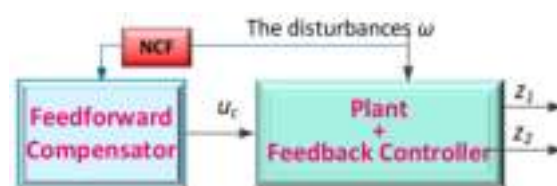


Figure 2. Block diagram of the disturbance compensator.

Accordingly, Theorem 2 presents a dynamic feedforward disturbance compensator design according to the dilated LMI in Lemma 3.

Theorem 2. Let a stable LPV system be in (1) for all admissible trajectories of parameters whose set is as in (2). If there exist a real scalar $\mu > 0$, the positive definite differentiable symmetric matrix $Y(\alpha)$ and the compatible matrices M , $N(\alpha)$, then the LMIs in (22) and (23) with the definitions in (24) hold for $\forall(\alpha) \in \Delta_R$ and $\forall(\alpha) \in \Delta_D$. In such a case, a dynamic H_∞ feedforward compensator in (21) is obtained from (25).

$$\begin{pmatrix} \Psi_{11}(\alpha) & \Psi_{12}(\alpha) & \Psi_{13}(\alpha) & \Psi_{14}(\alpha) \\ * & \Psi_{22}(\alpha) & \Psi_{23}(\alpha) & \Psi_{24}(\alpha) \\ * & * & \Psi_{33}(\alpha) & \Psi_{34} \\ * & * & * & \Psi_{44} \end{pmatrix} \prec 0 \quad (22)$$

$$Y(\alpha) = \begin{pmatrix} Y_1(\alpha) & Y_2(\alpha) \\ * & Y_3(\alpha) \end{pmatrix} \succ 0 \quad (23)$$

$$\begin{aligned} \Psi_{11}(\alpha) &= \begin{pmatrix} \dot{Y}_1(\alpha) - 2\mu Y_1(\alpha) & \dot{Y}_2(\alpha) - 2\mu Y_2(\alpha) \\ * & \dot{Y}_3(\alpha) - 2\mu Y_3(\alpha) \end{pmatrix}, \\ \Psi_{12}(\alpha) &= \begin{pmatrix} 0 & Y_1(\alpha) + R_1^T A^T(\alpha) + N^T(\alpha) B_2^T + \mu R_1^T \\ 0 & Y_1(\alpha) + R_1^T A^T(\alpha) + N^T(\alpha) B_2^T + \mu R_2^T(\alpha) \end{pmatrix}, \\ \Psi_{13}(\alpha) &= \begin{pmatrix} Y_2(\alpha) + M^T(\alpha) & R_1^T C_1^T(\alpha) + N^T(\alpha) D_{12}^T \\ Y_3(\alpha) + M^T(\alpha) & R_1^T C_1^T(\alpha) + N^T(\alpha) D_{12}^T \end{pmatrix}, \\ \Psi_{14}(\alpha) &= \begin{pmatrix} R_1^T C_2^T(\alpha) + N^T(\alpha) D_{22}^T \\ R_1^T C_2^T(\alpha) + N^T(\alpha) D_{22}^T \end{pmatrix}, \Psi_{22}(\alpha) = \begin{pmatrix} -\gamma^2 I & B_1(\alpha) + D_F^T(\alpha) B_2^T \\ * & -R_1 - R_1^T \end{pmatrix}, \\ \Psi_{23}(\alpha) &= \begin{pmatrix} B_F^T(\alpha) & D_{11}^T(\alpha) + D_F^T(\alpha) D_{12}^T \\ -R_1 - R_2^T(\alpha) & 0 \end{pmatrix}, \Psi_{24}(\alpha) = \begin{pmatrix} D_{21}^T(\alpha) + D_F^T(\alpha) D_{22}^T \\ 0 \end{pmatrix}, \\ \Psi_{33}(\alpha) &= \begin{pmatrix} -R_2(\alpha) - R_2^T(\alpha) & 0 \\ * & -I \end{pmatrix}, \Psi_{34} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Psi_{44} = -I \end{aligned} \quad (24)$$

$$\begin{aligned} A_c &= M(\alpha) R_2^{-1}(\alpha), B_c(\alpha) = B_F(\alpha), \\ C_c &= N(\alpha) R_2^{-1}(\alpha), D_c = D_F(\alpha) \end{aligned} \quad (25)$$

Proof of Theorem 2. The LMI in (8) is multiplied by the following matrix and the transpose of the matrix from the left–right side, respectively when we apply the congruence transformation.

$$\begin{pmatrix} Q^{-1} & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & Q^{-1} & 0 \\ 0 & 0 & 0 & I \end{pmatrix}$$

Therefore, the following LMI is acquired under the definitions in (27).

$$\begin{pmatrix} \dot{Y}(\alpha) - 2\mu Y(\alpha) & * & * & * \\ 0 & -\gamma^2 I & * & * \\ Y(\alpha) + A(\alpha)R + \mu R & B(\alpha) & -R - R^T & * \\ C(\alpha)R & D(\alpha) & 0 & -I \end{pmatrix} \prec 0 \quad (26)$$

$$Y(\alpha) = Q^{-T} P(\alpha) Q^{-1}, \dot{Y}(\alpha) = Q^{-T} \dot{P}(\alpha) Q^{-1}, R = Q^{-1} \quad (27)$$

The state space form from $\omega(t)$ to $Z(t)$ is as in (28) when the compensator in (21) is realized to (1). Here, the vector Z represents the outputs z and y , the vector X represents the states x in (1) and x_c in (21) and ω is the disturbance signal.

$$\begin{pmatrix} \dot{X}(t) \\ Z(t) \end{pmatrix} = \begin{pmatrix} A_{cl}(\alpha) & B_{cl}(\alpha) \\ C_{cl}(\alpha) & D_{cl}(\alpha) \end{pmatrix} \begin{pmatrix} x(t) \\ \omega(t) \end{pmatrix} \quad (28)$$

Accordingly, the state space matrices in (28) are obtained as in (29) with the realization. Therefore, the optimal controller is achieved by minimizing the L_2 gain from ω to Z .

$$\begin{aligned} A_{cl}(\alpha) &= \begin{pmatrix} A(\alpha) & B_2 C_c(\alpha) \\ 0 & A_c \end{pmatrix}, B_{cl}(\alpha) = \begin{pmatrix} B_1(\alpha) + B_2 D_c(\alpha) \\ B_c \end{pmatrix}, \\ C_{cl}(\alpha) &= \begin{pmatrix} C_1(\alpha) & D_{12} C_c(\alpha) \\ C_2(\alpha) & D_{22} C_c(\alpha) \end{pmatrix}, D_{cl}(\alpha) = \begin{pmatrix} D_{11}(\alpha) + D_{12} D_c(\alpha) \\ D_{21} + D_{22} D_c(\alpha) \end{pmatrix} \end{aligned} \quad (29)$$

Let us define the Lyapunov matrix as in (23), and the $R(\xi)$ matrix is defined as follows.

$$R(\alpha) = \begin{pmatrix} R_1 & R_1 \\ R_2(\alpha) & R_2(\alpha) \end{pmatrix}$$

Accordingly, if Lyapunov matrix $Y(\alpha)$ in (23), its derivative, the $R(\alpha)$ matrix and the state space matrices in (29) are placed in (26) in place of $A(\alpha)$, $B(\alpha)$, $C(\alpha)$ and $D(\alpha)$, we obtain a matrix inequality. If some expressions are defined as in (30), we obtain the LMI in (22) with the definitions in (24) and the proof is complete. $\square$

$$\begin{aligned} M(\alpha) &= A_c(\alpha)R_2(\alpha), \quad N = C_c(\alpha)R_2(\alpha), \\ B_F(\alpha) &= B_c(\alpha), \quad D_F(\alpha) = D_c(\alpha) \end{aligned} \quad (30)$$

Remark 4. The proposed LMIs provide four significant advantages. Some are mentioned in Remark 2. Primarily, the Lyapunov matrix can be designated in the LPV form, which reduces conservatism, without additional approaches such as grid, s-procedure, multi-convexity and the conditions in [17,18]. For example, many LMIs exist in [45] since multi-convexity conditions are used because of these products. In addition, the LMIs are based on the common BRL in (6), so LMIs are somewhat conservative because of the multiplication of the Lyapunov matrix. However, the proposed extended BRL is the best among those available, as shown in Table 1. Second, because of the products of system matrices with a Lyapunov matrix as in (6), infinite LMIs and nonconvex optimization occur, if the Lyapunov matrix can be in the LPV form. However, such a case does not occur for the proposed LMI formulation in (8). Thus, it is clear that the proposed solution is simpler. Third, the LPV controllers designed by PDLF as in [45,50] are generally dependent on the derivative of the Lyapunov matrix (matrix $A_c(\alpha)$ of the controller). However, the derivative operation causes noise problem in real-time practical applications. The proposed designs are not dependent on the derivative of the Lyapunov matrix. Finally, the controller matrices do not depend on the Lyapunov matrix, which is restricted by the symmetric condition in common designs. Instead, the matrices depend on a full matrix in the proposed designs. Moreover, it has a simple structure.

Remark 5. The proposed compensator does not include a feedback controller. It is well known that a feedforward method is applied to improve the reference tracking performance or to drop the disturbance or noise, and it does not affect stability, thus a feedback controller is needed. However, a feedback design is not needed for some plants, such as many mechanical and electrical systems, since they are stable systems. Therefore, the compensator can also be used for the stable systems with disturbances in addition to stabilized systems with a feedback controller. In addition, it can be used to improve the reference tracking by exchanging reference signals instead of disturbance inputs for the stabilized systems with a feedback controller. Accordingly, it is assumed that the systems are stable or stabilized with a feedback controller for all the examples throughout the paper. However, the internal stability of the LPV feedforward controller/compensator should be provided for all time-varying parameters, as in the above theorems. Otherwise, the stable system becomes unstable. This stability is achieved by decomposing the Lyapunov matrix without the need for any extra stability method for all the proposed theorems.

5. Simulation Studies

In this section, a numerical example has been realized regarding compensator design. All numerical optimization computations have been performed by Yalmip version R20210331 [47] and Sedumi version 1.30 [48], which have been used on a PC (2 cores 2.8 GHz CPU with 4 GB Ram).

Example 3. This example shows the LPV compensator design for the randomly generated LPV system in (31). The time-varying parameters $\alpha_1(t)$ and $\alpha_2(t)$ are bounded by (32). Although it normally appears as an indeterminate linear system, it is considered an LPV system by assuming

that the measurable or computable parameters are states of the nonlinear system. An LPV system is derived from a nonlinear system, unlike an uncertain linear system.

$$\begin{aligned}\dot{x}(t) &= \begin{pmatrix} 1 + \alpha_1(t) & -3 - \alpha_1(t) \\ 6 & -4 - \alpha_2(t) \end{pmatrix} x(t) + \begin{pmatrix} 0.5 + \alpha_1(t) \\ 0.8 \end{pmatrix} \omega(t) + \begin{pmatrix} 0.2 \\ 0.1 \end{pmatrix} u(t) \\ z(t) &= \begin{pmatrix} 1 & 0.1 \end{pmatrix} x(t) + (0.01)\omega(t) \\ y(t) &= \begin{pmatrix} 1 & 0 \end{pmatrix} x(t)\end{aligned}\quad (31)$$

$$\begin{aligned}0.1 \leq \alpha_1(t) \leq 0.7 \text{ with } -0.445 \leq \dot{\alpha}_1(t) \leq 1.82 \\ 1 \leq \alpha_2(t) \leq 5 \text{ with } -11.7 \leq \dot{\alpha}_2(t) \leq 2.295\end{aligned}\quad (32)$$

When the system is analyzed in view of time-varying parameters, the results in Figures 3 and 4 are obtained. The different time-varying parameters are tested for analysis in Figure 3. However, one of them is used for simulation which is in Figure 3b. Figure 3a is only used for analysis (Bode diagram and pole mapping) because the parameter change is assumed to be linear. Typically, both parameters are unlikely to vary linearly. The parameter change in Figure 3b occurs during the simulation. The aim is to observe the reactions to different parametric changes. For the analysis, the Bode diagram of the system is shown in Figure 4 with a sampling time of 50 ms and a simulation sampling time of 100 μ s. Figure 4a shows the Bode diagram of the system assuming the time-varying parameters in Figure 3a, whereas Figure 4b shows the Bode diagram assuming the time-varying parameters in Figure 3b. The figure shows that the time-varying parameters strongly affect the nonlinearity of the system according to Bode diagrams of the outputs from the inputs. From the Bode diagrams, the disturbance is more effective at approximately 3–6 rad/s. Therefore, the disturbance test signals have a low frequency.

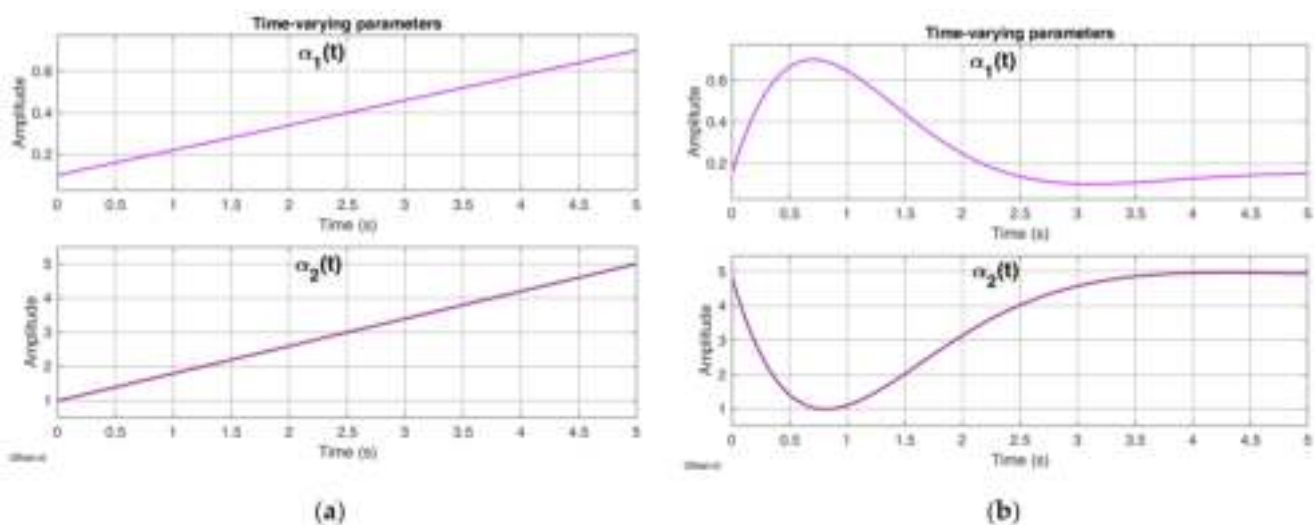


Figure 3. The time-varying parameters for: (a) Analysis; (b) Analysis and simulation.

Similarly, Figure 5 presents the pole map of the system for the various time-varying parameters in Figure 3b where the sampling time is 50 ms. From the pole map, two cases can be deduced. The poles of the whole transfer matrices of the system are in the left half of the plane. This case confirms the stability of the system in accordance with all the time-varying parameters mentioned in Remark 5. Otherwise, the compensator cannot be applied since it does not affect the stability. The other is that the parameters strongly affect the system.

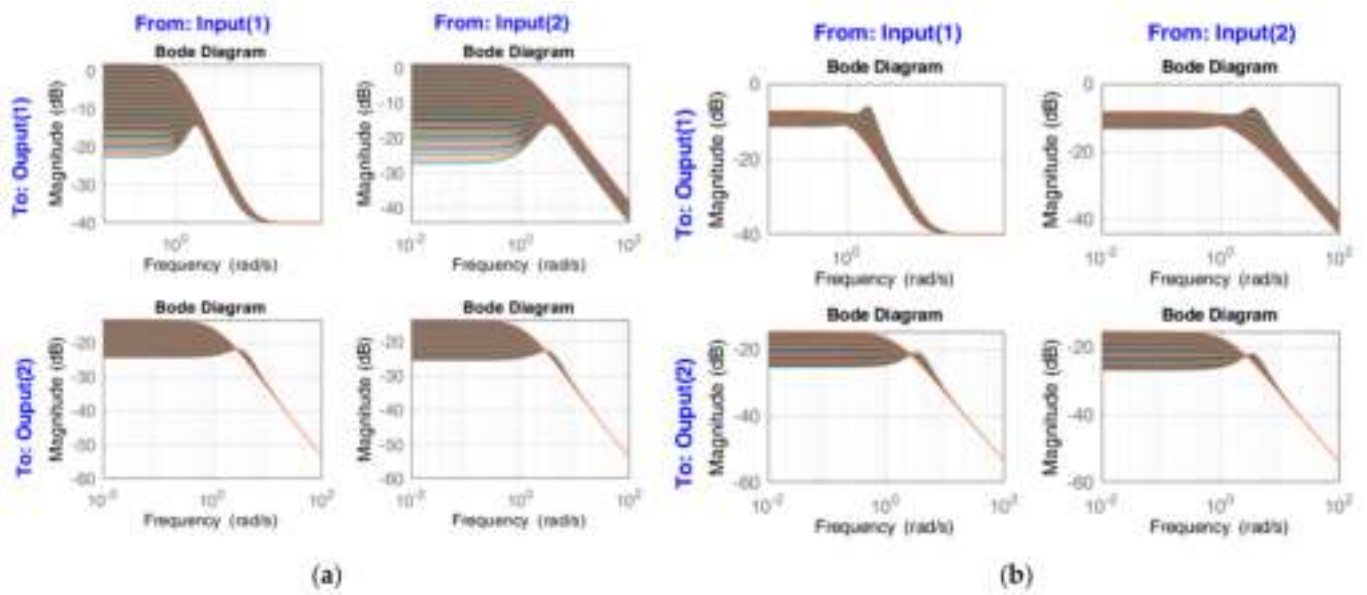


Figure 4. Bode diagrams of the system for time-varying parameters in: (a) Figure 3a; (b) Figure 3b.

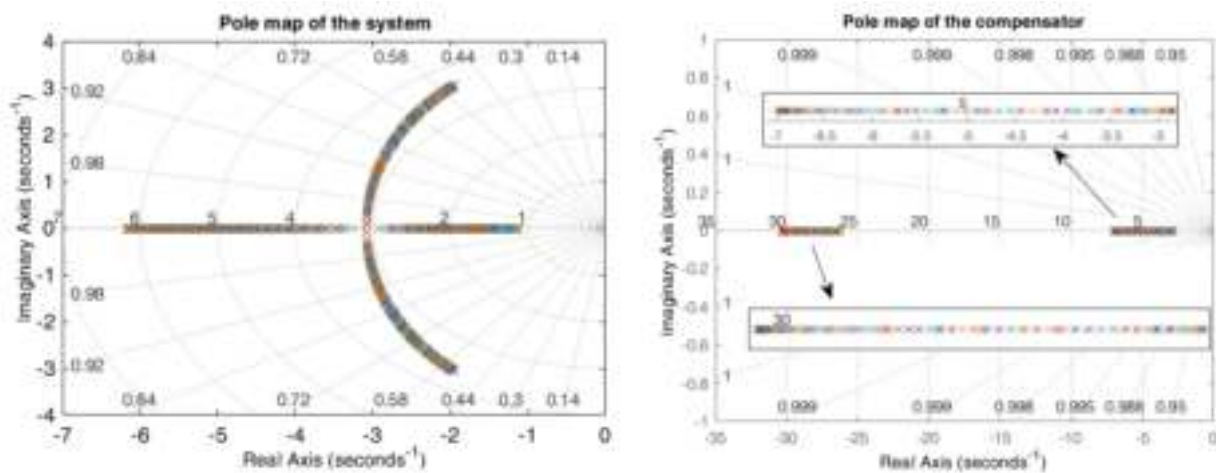


Figure 5. Pole maps of the system and compensator.

For the design, Figure 6 shows a schematic block diagram for the LPV compensator design where W_u and W_z represent the input and output weighting filters, respectively. However, an appropriate compensator is designed without these filters for this example. However, the filters can be used for reasons such as structural limitations as mentioned in Remark 6.

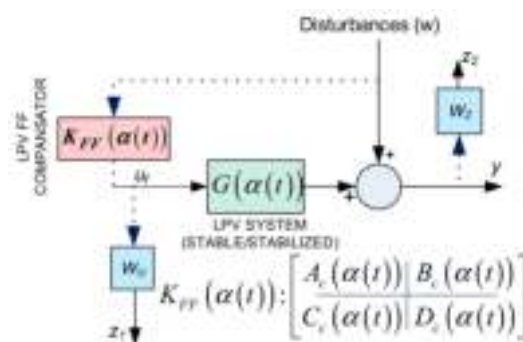


Figure 6. LPV compensator design block diagram.

Remark 6. The inputs and outputs are bounded because of the mechanical/electrical structure. Therefore, the constraints should be considered according to the problem of the related field. For example, the suspension deflection should not exceed the upper limit, whereas the tire deflection should be minimized as much as possible for active vehicle suspension systems. There are some LMI-based studies on the input and output constraints such as [51]. These are regulated with the NCF, as shown in Figure 2. NCF can be a coefficient or a filter such as a low-pass filter that is determined according to system behavior by considering the frequency and gain limits of inputs and outputs. The other way is the input and output weighting functions for the H_∞ controller design as shown in Figure 6 and the block diagram in [45]. Therefore, it can be decided for NCF considering the system structure. In particular, a filter for NCF can be needed at a high frequency instead of a fixed NC. This is because the compensator needs it to regulate the actuator limit at a high frequency.

The matrices are obtained in (33) when the LMIs in Theorem 2 are solved to minimize the γ value according to the parameter-dependent variables Y_i , B_{Fi} , D_{Fi} , M_i , N_i for $i = 0, 1, 2$ and the parameter-independent variables R_1 and R_2 . Therefore, the state space matrices of LPV compensator A_{ci} , B_{ci} , C_{ci} and D_{ci} in (21) for $i = 0, 1$ and 2 are affinely obtained from the matrices in (33) according to the time-varying parameters $\alpha_1(t)$ and $\alpha_2(t)$ in (32) for all t during the online operation. Hence, (34) shows the state space LPV controller matrices.

$$\begin{aligned} M_0 &= \begin{bmatrix} -163.81 & 12.724 \\ -1.7999 & -202.82 \end{bmatrix}, M_1 = \begin{bmatrix} 5.1074 & -22.914 \\ -11.194 & 36.474 \end{bmatrix}, \\ M_2 &= \begin{bmatrix} -10.704 & 16.627 \\ 15.009 & -84.506 \end{bmatrix}, R_1 = \begin{bmatrix} 7.1371 & -3.0487 \\ -20.361 & 92.818 \end{bmatrix}, \\ R_2 &= \begin{bmatrix} 7.139 & -3.0494 \\ -20.361 & 92.826 \end{bmatrix}, N_0 = \begin{bmatrix} -1175.1 & 1477.5 \end{bmatrix}, \\ N_1 &= \begin{bmatrix} -111.95 & 364.76 \end{bmatrix}, N_2 = \begin{bmatrix} -53.522 & 83.135 \end{bmatrix}, \\ B_{C0} &= \begin{bmatrix} -0.07727 \\ 0.51139 \end{bmatrix}, B_{C1} = \begin{bmatrix} 0.070224 \\ -0.46491 \end{bmatrix}, B_{C2} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ D_{C0} &= -3.3213, D_{C1} = -4.6489, D_{C2} = -0.0003355 \end{aligned} \quad (33)$$

$$\begin{aligned} A_c(\alpha_1, \alpha_2) &= \begin{bmatrix} -24.886 + 0.012541\alpha_1 - 1.0907\alpha_2 & -0.68047 - 0.24644\alpha_1 + 0.14329\alpha_2 \\ -7.1544 - 0.49361\alpha_1 - 0.54525\alpha_2 & -2.42 + 0.37671\alpha_1 - 0.92828\alpha_2 \end{bmatrix} \\ B_c(\alpha_1) &= \begin{bmatrix} -0.07727 + 0.070224\alpha_1 \\ 0.51139 - 0.46491\alpha_1 \end{bmatrix} \\ C_c(\alpha_1, \alpha_2) &= \begin{bmatrix} -131.53 - 4.9366\alpha_1 - 5.4537\alpha_2 & 11.596 + 3.7674\alpha_1 + 0.71644\alpha_2 \end{bmatrix} \\ D_c(\alpha_1, \alpha_2) &= -2.8863 - 4.6489\alpha_1 - 0.0003355\alpha_2 \end{aligned} \quad (34)$$

Accordingly, Figure 7a shows the Bode diagram of the compensator for the time-varying parameters in Figure 3a, whereas Figure 7b shows it for the time-varying parameters in Figure 3b for a sampling time of 50 ms. According to the pole maps and Bode diagrams, the LPV compensator significantly changes with the time-varying parameters.

The value of μ is determined to be 0.42 by line searching for a good response. Accordingly, the performance index γ value is 0.037268. This value clearly changes according to the system and input-output filters. It provides feasibility for very large time-varying parameters, as proven in state feedback controller design examples. Figure 8a also shows the Bode diagram of the system combined with the compensator from the disturbance input to the z and y outputs assuming the time-varying parameters in Figure 3a, whereas Figure 8b shows the Bode diagram assuming the time-varying parameters in Figure 3b. Figure 9 shows the applied disturbance test signals for the simulations. These are a bump signal as shown in Figure 9a; 5 rad/s sine signal shown in Figure 9b; and white noise signal shown in Figure 9c.

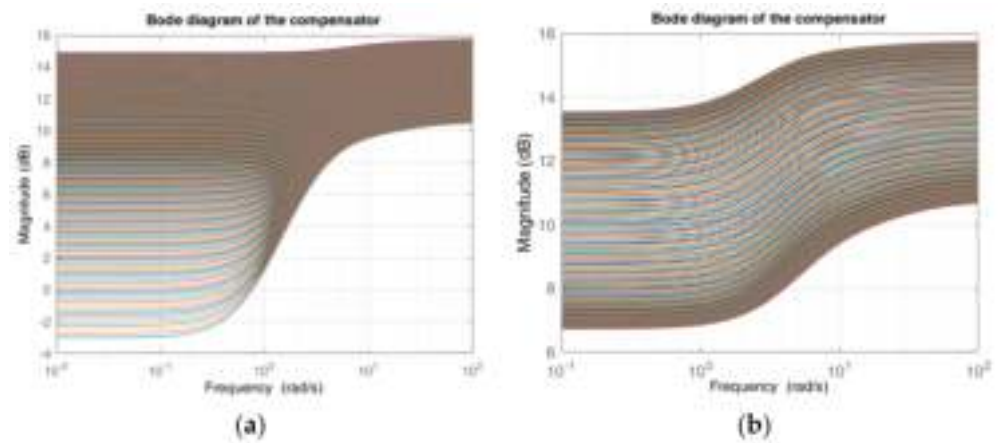


Figure 7. Bode diagrams of the compensator for time-varying parameters in: (a) Figure 3a; (b) Figure 3b.

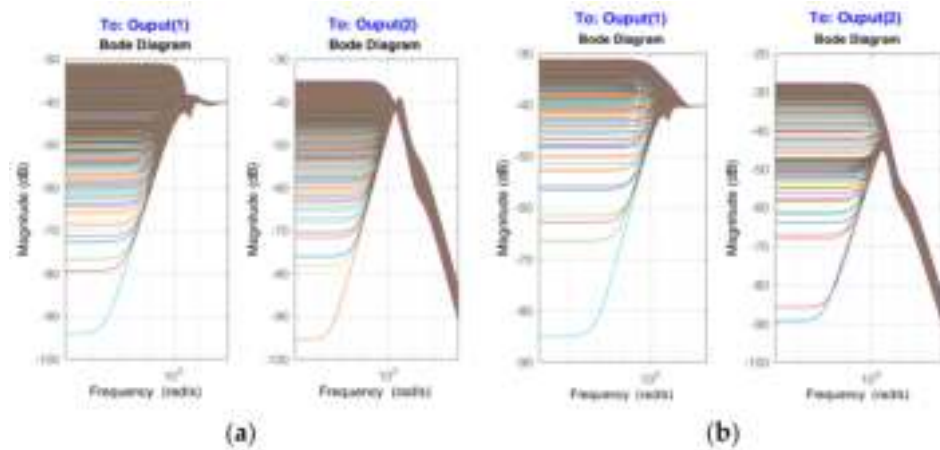


Figure 8. Bode diagrams of the system with the compensator for time-varying parameters in: (a) Figure 3a; (b) Figure 3b.

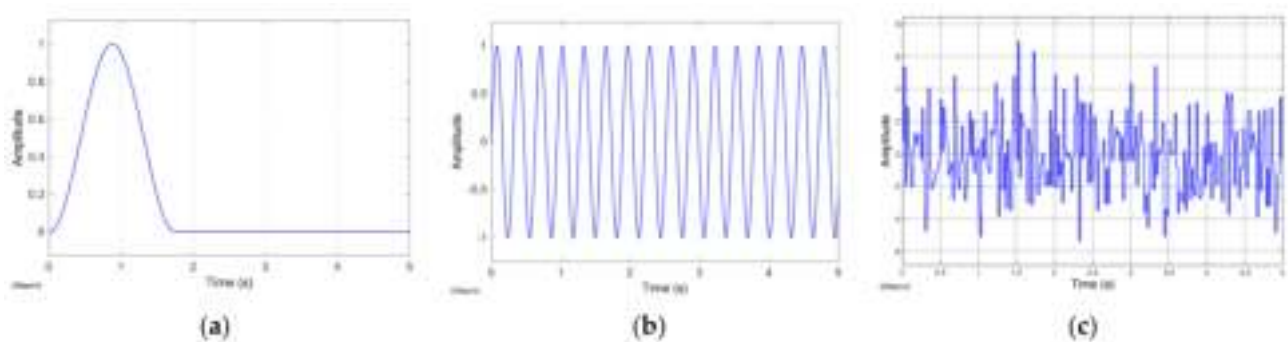


Figure 9. Applied test disturbances: (a) a bump; (b) a 5 rad/s sine signal; (c) a white noise signal.

As for the results, Figure 10 shows the outputs z and y for the external disturbance bump signal in Figure 9a. As seen from the outputs, the designed LPV compensator effectively attenuates the disturbance at a rate of approximately 100% compared with the case without the compensator.

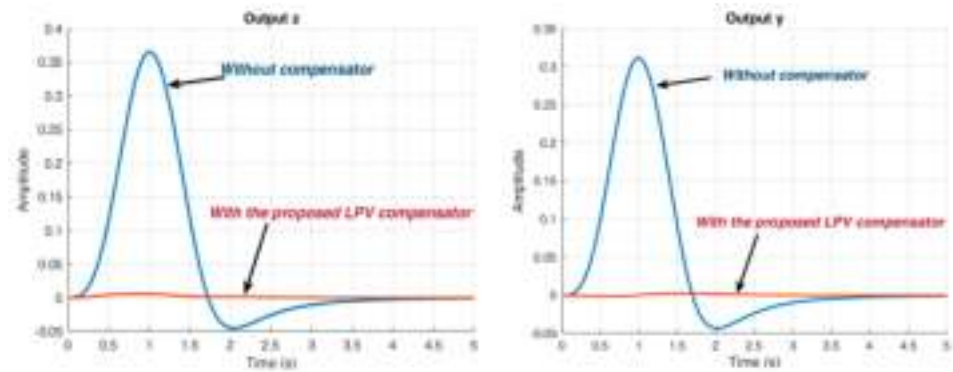


Figure 10. The outputs z and y for a bump disturbance signal.

Similarly, Figures 11 and 12 show the outputs z and y for the disturbance test signals in Figures 9b and 9c, respectively. The signals generated by the parameter-dependent compensator against three disturbances are shown in Figure 13.

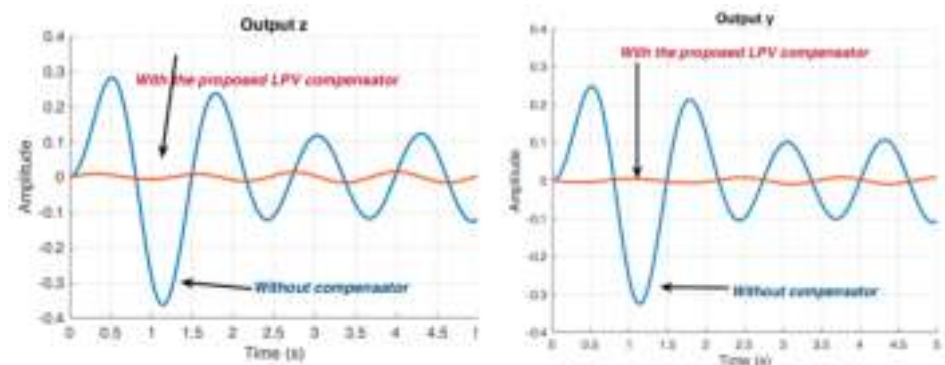


Figure 11. The outputs z and y for a 5 rad/s sinus disturbance.

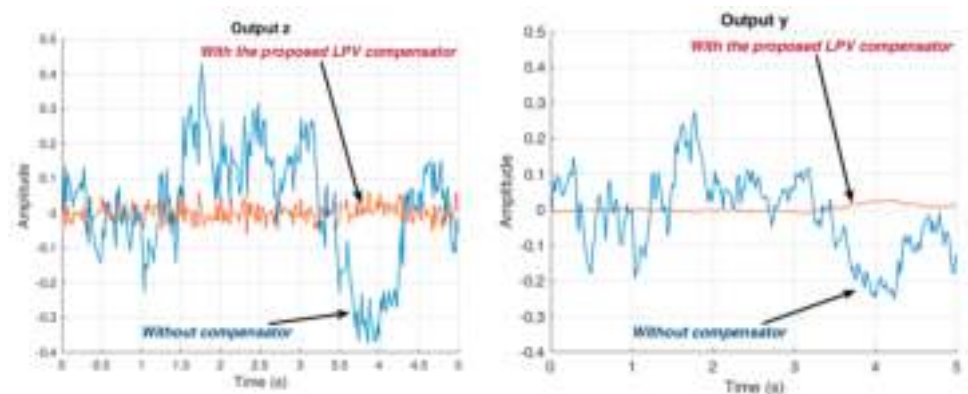


Figure 12. The outputs z and y for white noise disturbance.

The system with an LPV feedforward compensator is compared with the system without a compensator. It should be noted that the system must be a stable system or stabilized by a feedback controller. As seen from the outputs in Figures 11 and 12, the parameter-dependent compensator is quite successful, similar to the results in Figure 10. Thus, the effects of the external disturbances are reduced almost 100%. Therefore, the designed LPV feedforward compensator has significant performance against the disturbances. Finally, the compensator design is quite successful for the LPV system. As a result, the proposed compensator is quite successful for the rejection of measurable or observable disturbances.

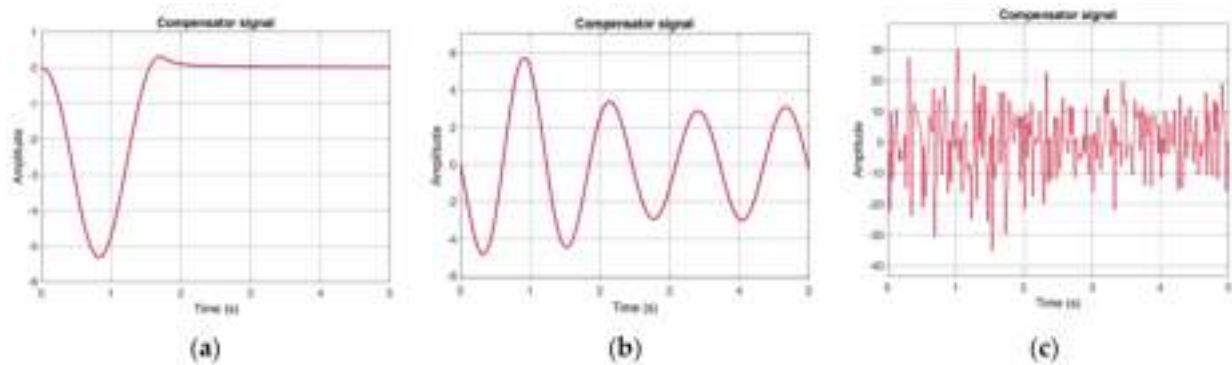


Figure 13. The generated force signals for: (a) a bump signal; (b) a 5 rad/s sine signal; (c) a white noise signal.

In the design of the proposed LPV compensator based on the dilated LMI, the Lyapunov matrix can be designated in the LPV form, which reduces conservatism, without additional approaches such as grid, s-procedure, multi-convexity in [11] and the conditions in [18,19]. For example, many LMI conditions exist for the LPV controller design in [45] since multi-convexity conditions are used because of the products $A(\alpha) \times P(\alpha)$ in (6). Therefore, the proposed solution is simpler and less conservative. Otherwise, either inefficient bilinear matrix inequality (BMI) solvers or iterative LMI algorithms are needed. The solutions do not require any iterative LMI or BMI algorithms.

6. Conclusions and Future Works

This paper focuses on a feedforward compensator against disturbances in addition to a state feedback controller for the LPV system. The Lyapunov matrix is divided from the matrices of system by injecting a slack real variable in the LMI-based designs; thus, the PDLM is designated. Therefore, the proposed LMI formulations for both state feedback controller and feedforward compensator are finite-dimensional and do not need extra conditions such as the s-procedure, grid and multi-convexity even if the PDLM is used. Thus, they have a simpler structure and lower conservativeness than similar state feedback and feedforward designs do. The numerical results of the state feedback controller show that the solution is less conservative and simpler even for very large parameter ranges. Especially for the compensator, the simulation results show that both the parameter-dependent and parameter-independent feedforward compensators have significant performance against disturbances because the effects of the external disturbances are almost completely eliminated. Therefore, the validity and advantages of the proposed LMIs are effectively obtained with numerical and simulation examples. As a result, the proposed LMI solutions are better for the systems in terms of simplicity and conservativeness.

For further studies, the first of these can be considered as obtaining the optimum real scalar μ via metaheuristic optimization methods. The second can design both feedback control and a feedforward compensator at the same time for unstable systems. The third is the design of switching the LPV feedforward compensator via the dilated LMI. Furthermore, this proposed feedforward compensator with a switched LPV feedforward compensator can be compared with respect to criteria such as performance and complexity.

Author Contributions: Conceptualization, M.İ. and Y.A.; methodology, M.İ. and Y.A.; software, M.İ. and Y.A.; validation, Y.A.; formal analysis, Y.A.; investigation, M.İ. and Y.A.; resources, M.İ.; data curation, M.İ.; writing—original draft preparation, M.İ.; writing—review and editing, Y.A.; visualization, M.İ. and Y.A.; supervision, Y.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: Author Mustafa İnci was employed by the company Volta Motor Industry and Trade Inc. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

- Kim, C. Robust H_∞ Static Output Feedback Control for TCP/AQM Routers Based on LMI Optimization. *Electronics* **2024**, *13*, 2165. [\[CrossRef\]](#)
- Kim, Y.; Tran, T.V.; Kim, K.-H. Robust EMPC-Based Frequency-Adaptive Grid Voltage Sensorless Control for an LCL-Filtered Grid-Connected Inverter. *Electronics* **2024**, *13*, 998. [\[CrossRef\]](#)
- Németh, Z.; Kuczmann, M. Linear-Matrix-Inequality-Based Controller and Observer Design for Induction Machine. *Electronics* **2022**, *11*, 3894. [\[CrossRef\]](#)
- Ndabarushimana, E.; Qin, N.; Ma, L. Disturbance Decoupling for a Single-Phase Pulse Width Modulation Rectifier Based on an Extended H-Infinity Filter. *Electronics* **2023**, *12*, 2765. [\[CrossRef\]](#)
- Boyd, S.; El Ghaoui, L.; Feron, E.; Balakrishnan, V. *Linear Matrix Inequalities in System and Control Theory*; Society for Industrial and Applied Mathematics (SIAM) Publishing: Philadelphia, PA, USA, 1994; Volume 15, ISBN 0898714850.
- De Oliveira, M.C.; Geromel, J.C.; Bernussou, J. Extended H_2 and H Norm Characterizations and Controller Parametrizations for Discrete-Time Systems. *Int. J. Control* **2002**, *75*, 666–679. [\[CrossRef\]](#)
- Bara, G.I. Robust Analysis and Control of Parameter-Dependent Uncertain Descriptor Systems. *Syst. Control Lett.* **2011**, *60*, 356–364. [\[CrossRef\]](#)
- Cao, Y.-Y.; Lin, Z. A Descriptor System Approach to Robust Stability Analysis and Controller Synthesis. *IEEE Trans. Autom. Control* **2004**, *49*, 2081–2084. [\[CrossRef\]](#)
- De Caigny, J.; Camino, J.F.; Oliveira, R.C.L.F.; Peres, P.L.D.; Swevers, J. Gain-Scheduled H_2 and H_∞ Control of Discrete-Time Polytopic Time-Varying Systems. *IET Control Theory Appl.* **2010**, *4*, 362–380. [\[CrossRef\]](#)
- Oliveira, M.C.D.; Geromel, J.C.; Bernussou, J. An LMI Optimization Approach to Multiobjective Controller Design for Discrete-Time Systems. In Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304), Phoenix, AZ, USA, 7–10 December 1999; IEEE: Piscataway, NJ, USA, 1999; Volume 4, pp. 3611–3616.
- Gahinet, P.; Apkarian, P.; Chilali, M. Affine Parameter-Dependent Lyapunov Functions and Real Parametric Uncertainty. *IEEE Trans. Automat. Control* **1996**, *41*, 436–442. [\[CrossRef\]](#)
- Pipeleers, G.; Demeulenaere, B.; Swevers, J.; Vandenbergh, L. Systems & Control Letters Extended LMI Characterizations for Stability and Performance of Linear Systems. *Syst. Control Lett.* **2009**, *58*, 510–518. [\[CrossRef\]](#)
- Ku, C.-C.; Chen, G.-W. Gain-Scheduled Controller Design for Discrete-Time Linear Parameter Varying Systems with Multiplicative Noises. *Int. J. Control Autom. Syst.* **2015**, *13*, 1382–1390. [\[CrossRef\]](#)
- Becker, G.; Packard, A. Robust Performance of Linear Parametrically Varying Systems Using Parametrically-Dependent Linear Feedback. *Syst. Control Lett.* **1994**, *23*, 205–215. [\[CrossRef\]](#)
- Feron, E.; Apkarian, P.; Gahinet, P. S-Procedure for the Analysis of Control Systems with Parametric Uncertainties via Parameter-Dependent Lyapunov Functions. In Proceedings of the 1995 American Control Conference, Seattle, WA, USA, 21–23 June 1995; IEEE: Piscataway, NJ, USA, 1995; Volume 1, pp. 968–972.
- Wei, X.; Verhaegen, M. LMI Solutions to the Mixed H_-/H_∞ Fault Detection Observer Design for Linear Parameter-Varying Systems. *Int. J. Adapt. Control Signal Process.* **2011**, *25*, 114–136. [\[CrossRef\]](#)
- Lim, S. *Analysis and Control of Linear-Parameter-Varying Systems*; Stanford University: Stanford, CA, USA, 1999; ISBN 0599241535.
- Oliveira, R.C.L.F.; Peres, P.L.D. Parameter-Dependent LMIs in Robust Analysis: Characterization of Homogeneous Polynomially Parameter-Dependent Solutions via LMI Relaxations. *IEEE Trans. Autom. Control* **2007**, *52*, 1334–1340. [\[CrossRef\]](#)
- Ebihara, Y.; Peaucelle, D.; Arzelier, D. *S-Variable Approach to LMI-Based Robust Control*; Springer: Berlin/Heidelberg, Germany, 2015; ISBN 978-1-4471-6605-4.
- Ebihara, Y.; Hagiwara, T. New Dilated LMI Characterizations for Continuous-Time Multiobjective Controller Synthesis. *Automatica* **2004**, *40*, 2003–2009. [\[CrossRef\]](#)
- Song, L.; Yang, J. An Improved Approach to Robust Stability Analysis and Controller Synthesis for LPV Systems. *Int. J. Robust Nonlinear Control* **2010**, *21*, 1574–1586. [\[CrossRef\]](#)
- Apkarian, P.; Tuan, H.D.; Bernussou, J. Continuous-Time Analysis, Eigenstructure Assignment, and H_2 Synthesis with Enhanced Linear Matrix Inequalities (LMI) Characterizations. *IEEE Trans. Autom. Control* **2001**, *46*, 1941–1946. [\[CrossRef\]](#)
- Ebihara, Y.; Hagiwara, T. A Dilated LMI Approach to Robust Performance Analysis of Linear Time-Invariant Uncertain Systems. *Automatica* **2005**, *41*, 1933–1941. [\[CrossRef\]](#)
- Trofino, A.; Coutinho, D.F.; Barbosa, K.A. Improved H_2 and H_∞ Conditions for Robust Analysis and Control Synthesis of Linear Systems. *Control Autom.* **2005**, *16*, 427–434. [\[CrossRef\]](#)
- Shaked, U. Improved LMI Representations for the Analysis and the Design of Continuous-Time Systems with Polytopic Type Uncertainty. *IEEE Trans. Autom. Control* **2001**, *46*, 652–656. [\[CrossRef\]](#)
- Wei, J.; Lee, L. Further Improvement on LMI Representations for the Analysis and Design of Continuous-Time Systems with Polytopic Type Uncertainty 1. In Proceedings of the 2004 5th Asian Control Conference (IEEE Cat. No. 04EX904), Melbourne, VIC, Australia, 20–23 July 2004; Volume 2, pp. 1–7.

27. Sajjadi-Kia, S.; Jabbari, F. On Bounded Real Matrix Inequality Dilation. *Int. J. Control* **2012**, *85*, 1593–1601. [\[CrossRef\]](#)
28. Xie, W. An Equivalent LMI Representation of Bounded Real Lemma for Continuous-Time Systems. *J. Inequalities Appl.* **2008**, *2008*, 672905. [\[CrossRef\]](#)
29. Dabboussi, K.; Zrida, J. A Dilated LMI Approach to Robust H_∞ Static State Feedback Controller Synthesis. In Proceedings of the 2009 16th IEEE International Conference on Electronics, Circuits and Systems—(ICECS 2009), Yasmine Hammamet, Tunisia, 13–16 December 2009; pp. 491–494.
30. Rodrigues, L.A.; Oliveira, R.C.L.F.; Camino, J.F. Parameterized LMIs for Robust and State Feedback Control of Continuous-Time Polytopic Systems. *Int. J. Robust Nonlinear Control* **2018**, *28*, 940–952. [\[CrossRef\]](#)
31. Hu, C.; Jaimoukha, I.M. New Iterative Linear Matrix Inequality Based Procedure for H_2 and H_∞ State Feedback Control of Continuous-Time Polytopic Systems. *Int. J. Robust Nonlinear Control* **2021**, *31*, 51–68. [\[CrossRef\]](#)
32. Guzmán, J.L.; Häggglund, T. Tuning Rules for Feedforward Control from Measurable Disturbances Combined with PID Control: A Review. *Int. J. Control* **2021**, *97*, 2–15. [\[CrossRef\]](#)
33. Zhu, W.; Zhang, Z.; Chen, J.; Zhao, S.; Huang, S. Dynamic Data Reconciliation to Enhance the Performance of Feedforward/Feedback Control Systems with Measurement Noise. *J. Process Control* **2021**, *108*, 12–24. [\[CrossRef\]](#)
34. García-Mañas, F.; Guzmán, J.L.; Rodríguez, F.; Berenguel, M.; Häggglund, T. Experimental Evaluation of Feedforward Tuning Rules. *Control Eng. Pract.* **2021**, *114*, 104877. [\[CrossRef\]](#)
35. Du, Y.; Cao, W.; She, J.; Wu, M.; Fang, M. Disturbance Rejection via Feedforward Compensation Using an Enhanced Equivalent-Input-Disturbance Approach. *J. Frankl. Inst.* **2020**, *357*, 10977–10996. [\[CrossRef\]](#)
36. Huang, M.; Deng, Y.; Li, H.; Wang, J. Torque Ripple Attenuation of PMSM Using Improved Robust Two-Degree-of-Freedom Controller via Extended Sliding Mode Parameter Observer. *ISA Trans.* **2022**, *129*, 558–571. [\[CrossRef\]](#)
37. Adam, E.J.; Marchetti, J.L. Designing and Tuning Robust Feedforward Controllers. *Comput. Chem. Eng.* **2004**, *28*, 1899–1911. [\[CrossRef\]](#)
38. Peng, C.; Xu, H.; Zhang, J. Two Degree-of-Freedom Control Design with Improved H_∞ LMI Representation. *J. Inequalities Appl.* **2012**, *2012*, 147. [\[CrossRef\]](#)
39. Köse, I.E.; Scherer, C.W. Robust $\mathcal{L}_2$ -Gain Feedforward Control of Uncertain Systems Using Dynamic IQCs. *Int. J. Robust Nonlinear Control* **2009**, *19*, 1224–1247. [\[CrossRef\]](#)
40. Alma, M.; Martinez, J.J.; Landau, I.D.; Buche, G. Design and Tuning of Reduced Order H-Infinity Feedforward Compensators for Active Vibration Control. *IEEE Trans. Control Syst. Technol.* **2012**, *20*, 554–561. [\[CrossRef\]](#)
41. Peng, C.; Zhang, Z.; Zou, J.; Li, K.; Zhang, J. Internal Model Based Robust Inversion Feedforward and Feedback 2DOF Control for LPV System with Disturbance. *J. Process Control* **2013**, *23*, 1415–1425. [\[CrossRef\]](#)
42. Yuan, C.; Wu, F. Robust H_2 and H_∞ Switched Feedforward Control of Uncertain LFT Systems. *Int. J. Robust Nonlinear Control* **2016**, *26*, 1841–1856. [\[CrossRef\]](#)
43. Fleps-Dezasse, M.; Bünte, T.; Svaricek, F.; Brembeck, J. LPV Feedforward Control of Semi-Active Suspensions for Improved Roll Stability. *Control Eng. Pract.* **2018**, *78*, 1–11. [\[CrossRef\]](#)
44. Altun, Y.; Gulez, K.; Mumcu, T.V. Static LPV Feedforward Controller Synthesis for Linear Parameter Varying Systems. In Proceedings of the 2013 9th Asian Control Conference (ASCC), Istanbul, Turkey, 23–26 June 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 1–4.
45. Altun, Y.; Gulez, K. Linear Parameter Varying Feedforward Control Synthesis Using Parameter-Dependent Lyapunov Function. *Nonlinear Dyn.* **2014**, *78*, 2293–2307. [\[CrossRef\]](#)
46. Venkataraman, R.; Seiler, P. Convex LPV Synthesis of Estimators and Feedforwards Using Duality and Integral Quadratic Constraints. *Int. J. Robust Nonlinear Control* **2018**, *28*, 953–975. [\[CrossRef\]](#)
47. Löfberg, J. Automatic Robust Convex Programming. *Optim. Methods Softw.* **2012**, *27*, 115–129. [\[CrossRef\]](#)
48. Labit, Y.; Peaucelle, D.; Henrion, D. SEDUMI INTERFACE 1.02: A Tool for Solving LMI Problems with SEDUMI. In Proceedings of the 2002 IEEE International Symposium on Computer Aided Control System Design, CACSD 2002—Proceedings, Glasgow, UK, 20 September 2002; pp. 272–277.
49. Gahinet, P.; Nemirovskii, A.; Laub, A.J.; Chilali, M. LMI Control Toolbox. In Proceedings of the 1994 33rd IEEE Conference on Decision and Control, Lake Buena Vista, FL, USA, 14–16 December 1994; The Mathworks, Inc.: Natick, MA, USA, 1995; pp. 4–13.
50. Apkarian, P.; Adams, R.J. Advanced Gain-Scheduling Techniques for Uncertain Systems. *IEEE Trans. Control Syst. Technol.* **1998**, *6*, 21–32. [\[CrossRef\]](#)
51. Chen, H.; Guo, K.H. Constrained H_∞ Control of Active Suspensions: An LMI Approach. *IEEE Trans. Control. Syst. Technol.* **2005**, *13*, 412–421. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

An Adaptive High-Efficiency LED Backlight Driver

Xinyu Xu, Qiyue Zhuo, Chunhui Jiang, Zichao Zhou, Junlong Tang  and Wanghui Zou *

School of Physical and Electronic Sciences, Changsha University of Science and Technology,
Changsha 410114, China

* Correspondence: zouwh@csust.edu.cn

Abstract: An adaptive high-efficiency light-emitting Diode (LED) backlight driver scheme has been proposed to address the issue of additional power loss caused by LED forward voltage variation. In this scheme, the peak current and the duty cycle of each LED channel are adjusted separately through an adaptive control algorithm to minimize the voltage drop on the linear current regulator (LCR) of each LED channel to reduce the excessive power loss in each LED channel and enhance the total power efficiency. A linear current regulator, suitable for adaptive control, is designed on a 0.18 μm 5V complementary metal-oxide-semiconductor (CMOS) process. Simulation results demonstrate that the linear current regulator can achieve a linearly adjustable channel current ranging from 0 to 48 mA with a current resolution of 0.2 mA. Across different process corners and temperatures, the maximum error for the full current range is less than 0.1%. The core area of chip layout is about 0.1 mm^2 . The complete driver prototype comprises the LCR chips, external power MOS transistors, digital module, and LED chains. The test results show that the power loss of the linear current regulator has been significantly reduced, and the power efficiency of each LED channel has been measured at around 98.1%.

Keywords: LED backlight driver; adaptive control; linear current regulation; high efficiency



Citation: Xu, X.; Zhuo, Q.; Jiang, C.; Zhou, Z.; Tang, J.; Zou, W. An Adaptive High-Efficiency LED Backlight Driver. *Electronics* **2024**, *13*, 3057. <https://doi.org/10.3390/electronics13153057>

Academic Editor: Francesco Driussi

Received: 22 June 2024

Revised: 25 July 2024

Accepted: 30 July 2024

Published: 2 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Light-emitting diodes (LEDs) have been widely adopted as backlight sources for liquid crystal display (LCD) devices and other applications. Compared to traditional light sources, LEDs offer significant advantages such as high efficiency, excellent color stability, high reliability, long lifespan, compact size, and environmental safety [1–4]. Since the brightness of LED is proportional to its current, an efficient and accurate driver circuit is essential in LED backlight applications to achieve precise brightness control. Additionally, parallel LED channels (LED chains) are always employed to ensure sufficient coverage [5,6]. A traditional LED backlight driver is primarily composed of a boost converter, parallel LED channels, and corresponding linear current regulators (LCRs). The boost converter provides sufficient voltage to turn on all LEDs in each channel, and the LCR regulates the current for each channel to control brightness [7–9]. As an important area of power management application, the design of LED backlight drivers has always been a popular research focus for numerous universities and research institutions worldwide. The objective is to optimize system performance through innovative circuit designs and system architectures, with an emphasis on metrics such as system efficiency [10–15], dimming ratio [16–20], and current accuracy [21–25].

Due to process variations, aging effects, and temperature changes, LEDs of the same type can exhibit different forward voltage, resulting in significant voltage discrepancies across different LED channels. Consequently, it is necessary to introduce a certain amount of voltage headroom to ensure correct functionality under various working conditions. However, this also leads to increased power loss and deteriorated power efficiency [26,27]. A digital control mechanism based on an R-DAC was proposed in [28] to address the problem of increased power loss due to excessive voltage. A comparison between the voltage

on the LCR and a pre-set reference voltage is conducted, which drives the adjustment of the feedback voltage of the DC–DC boost converter via a digital-to-analog converter (DAC). The boost converter's output is then adjusted to maintain the LCR voltage at a fixed level. In [29], an adaptive reference tracking technique was proposed, using a “Minimum Voltage” circuit and a 3-bit adjustable resistor as a feedback resistor to adjust the output of the boost converter. The accuracy of the output voltage is influenced by the precision and adjustment range of the resistor. In [30–34], through “Minimum Voltage Selector” or “Lowest channel selector”, the minimum LCR voltage among channels is detected and used to dynamically adjust the duty cycle of the boost converter's switch signal and change the boost converter's output. In [35,36], diodes are used to select the minimum LCR voltage, a method that is not easily implemented in standard CMOS processes. In [2], the advantages and disadvantages of various LED driver topologies were analyzed, and future research directions were identified.

In general, for most existing solutions, the main strategy is to adjust the boost converter's output voltage to reduce the excessive voltage on the LCR, thereby reducing losses and improving efficiency. However, for multi-channel configuration, due to the inter-channel differences, it is more effective to optimize each channel separately to further reduce loss and enhance efficiency. In this paper, we propose an adaptive high-efficiency LED backlight driving scheme that addresses the issue of inter-channel differences. Both linear current regulation and pulse width modulation (PWM) are employed for each channel. Through linear current regulation, the channel current is adjusted, and the voltage on the LCR in each channel is minimized to a predefined fixed value. Subsequently, through pulse width modulation, the duty cycle of channel current is adjusted, calibrating the average current of each channel to the same level. This strategy ensures that each LED channel operates with minimal loss, thereby maximizing efficiency.

The structure of the remaining parts of this article is as follows: Section 2 elaborates on the adaptive control technology designed in this paper. Section 3 discusses and analyzes the simulation results and test results of the proposed adaptive high-efficiency LED backlight driver. Section 4 concludes the paper.

2. Adaptive Control Technique

2.1. Principle of Adaptive Control

Figure 1 illustrates the structure of the proposed adaptive high-efficiency LED backlight driver, which comprises a brightness controller, a boost converter, and several LED channels. Each LED channel consists of an LED chain, an LCR, and a high-voltage MOS transistor acting as a switch for duty cycle control. The brightness controller includes a 9-bit R-DAC to produce V_C , which serves as a voltage reference and determines the output of the boost converter V_{boost} . The SENSE signal indicates whether the LCR is on the right status. The brightness controller monitors SENSE of each channel and generates V_C , PWM, and CM_SW signals following a specific algorithm and steps. CM_SW and PWM are multi-bit digital signals that control the peak current and the duty cycle of the channel, respectively. This process ensures that each LED channel operates at a high-efficiency state.

From the figure, for channel n , there exists a voltage relationship as

$$V_{\text{boost}} = V_{\text{LED},n} + V_{\text{MP},n} + V_{\text{LCR},n} \quad (1)$$

where V_{LED} is the voltage across the LED chain or the summation of the forward voltage of each LED. V_{MP} is the source-drain voltage of the high-voltage MOS transistor MP, and V_{LCR} denotes the LCR voltage. The “ n ” in subscript indicates channel n , which applies to all symbols in this paper. If the adaptive control is disabled and the LED channel is not in the optimal state, there will be an excessive portion in the LCR voltage, which can be expressed as follows:

$$V_{\text{LCR},n} = V_{\text{LCR}_{\text{min}}} + V_{\text{LCR}_{\text{exc},n}} \quad (2)$$

where V_{LCR_min} is the minimum required LCR voltage to ensure proper operation, that is to say, the output transistor in the LCR works properly and maintains accurate output current. V_{LCR_exc} represents the excessive portion of the LCR voltage. For one channel, the power loss can be calculated by

$$P_{loss,n} = I_{ave,n} \times V_{loss,n} = I_{ave,n} \times (V_{LCR,n} + V_{MP,n}) \quad (3)$$

where $I_{ave,n}$ is the average current, which is determined by the peak current $I_{peak,n}$ and the current duty cycle δ_n . $V_{loss,n}$ is the summation of $V_{LCR,n}$ and $V_{MP,n}$. If the on-resistances of the high-voltage MOS transistors are low enough, V_{MP} has limited impact on P_{loss} . Because V_{MP} and V_{LCR_min} are prerequisite to maintain the MOS and LCR operate appropriately, the excessive power loss is defined as

$$P_{loss\_exc,n} = I_{ave,n} \times V_{LCR\_exc,n} \quad (4)$$

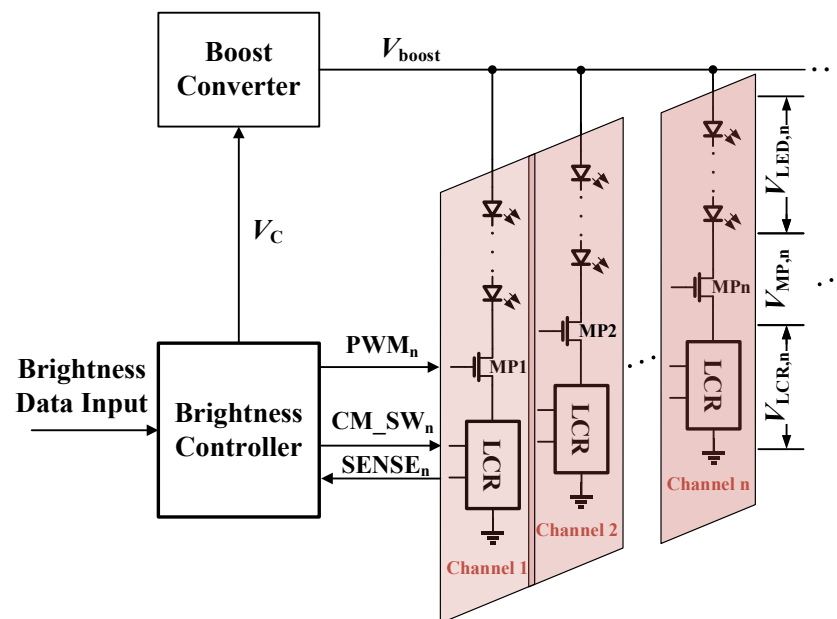


Figure 1. Structure of the proposed adaptive high-efficiency LED backlight driver.

2.2. Adaptive Control Process

An adaptive control process suitable for multi-channel LED drivers is designed to reduce excessive power loss caused by the excessive voltage. At system startup, the brightness controller initiates an initialization process, during which the peak current and duty cycle of each LED channel are individually set to achieve uniform brightness and simultaneously minimize excessive power loss. After the initialization phase, the brightness controller transitions into the tracking phase, where it continuously monitors the status of the LCR. The peak current and duty cycle settings are then adjusted accordingly to accommodate fluctuations in power supply, temperature, and other factors.

Figure 2 shows the flowchart of the initialization phase. Firstly, the brightness input data, including the nominal peak current I_{set} and the duty cycle δ_{set} , are loaded into the brightness controller. Subsequently, the LCR current of each channel is set to I_{set} via the CM_SW signal. The brightness controller then begins to increase V_C , causing V_{boost} to decrease accordingly. Meanwhile, the brightness controller simultaneously monitors the status signal SENSE from each channel. Once all SENSE signals change from 0 to 1, the increase in V_C stops. At this point, the peak currents of all channels have reached the preset value of I_{set} , and there is always one channel, assumed to be channel k , whose SENSE signal is the last to change.

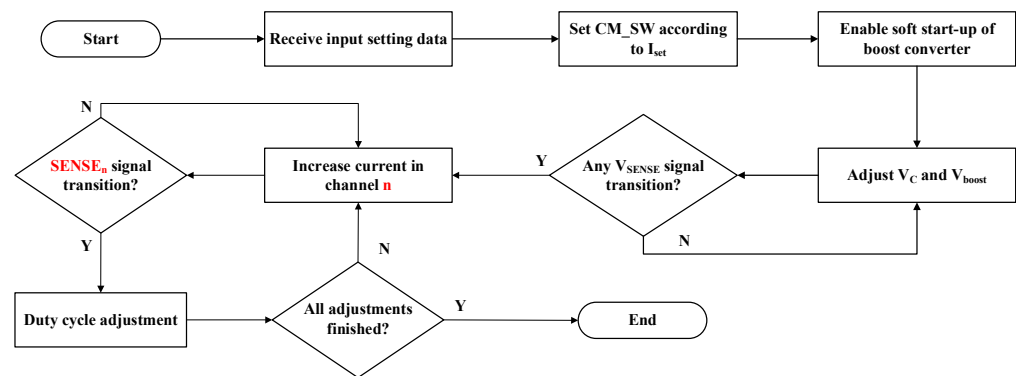


Figure 2. Flowchart of the adaptive control process in the initialization phase.

Next, a peak current adjustment step is carried out sequentially on all channels except channel k . The brightness controller attempts to increase the channel current by providing a new CM_SW setting. If $SENSE$ remains at 1, the brightness controller continues this process. However, if $SENSE$ changes from 1 to 0, the brightness controller reverts to the previous CM_SW setting, concluding the adjustment process for that channel. The corresponding channel then has a new peak current. The brightness controller proceeds to conduct the adjustment process for the subsequent channel. Once all current adjustments are completed, the voltage on each LCR approaches V_{LCR_min} , and the excess component has been minimized.

Finally, in order to ensure the same brightness, each LED channel needs to have the same average current. Therefore, the duty cycle of channel n (δ_n) is adjusted by the brightness controller as follows:

$$\delta_n = \delta_{set} \times \frac{I_{set}}{I_{peak,n}} \quad (5)$$

where $I_{peak,n}$ is the peak current of n -th LED channel after adjustment process as described above. δ_{set} and I_{set} represent the initial duty cycle and peak current, respectively. It is worth noting that, according to Equation (4), the average current remains unchanged.

Through the aforementioned process, the peak current $I_{peak,n}$ and duty cycle δ_n for each LED channel are adjusted, thereby minimizing additional power consumption resulting from excessive voltage. The adjusted average current closely aligns with the initially set average current. Given that the brightness output of LEDs correlates directly with their current, the adaptive control method ensures that the brightness levels across all LED channels remain consistent. Upon entering the tracking phase, the main principle resembles that of the initialization phase. Both of them combine the adjustment of peak current and duty cycle and will not be described in detail in this article.

2.3. Linear Current Regulator

Because LED brightness is directly influenced by its current, precise brightness control necessitates accurate current regulation. In many LED backlight driver solutions, an LCR is typically employed for generating and controlling the LED current. Figure 3 illustrates two typical LCR circuit configurations [37]. A key design objective for LCR circuits is achieving high current accuracy. In Figure 3a, the operational amplifier and the transistor M1 form a negative feedback, ensuring that V_S equals to V_{REF} . Consequently, the output current I_{LED} is V_{REF}/R_S . Clearly, the accuracy of R_S directly affects I_{LED} 's precision. It is important to note that the driving currents for backlight LEDs are generally substantial. In this paper, the maximum driving current is designed to be 48 mA. Additionally, to minimize power loss, V_S or V_{REF} should be kept as low as possible, necessitating a small R_S to achieve high currents. However, excessively small R_S can significantly enhance the negative impact of circuit parasitics on output current accuracy.

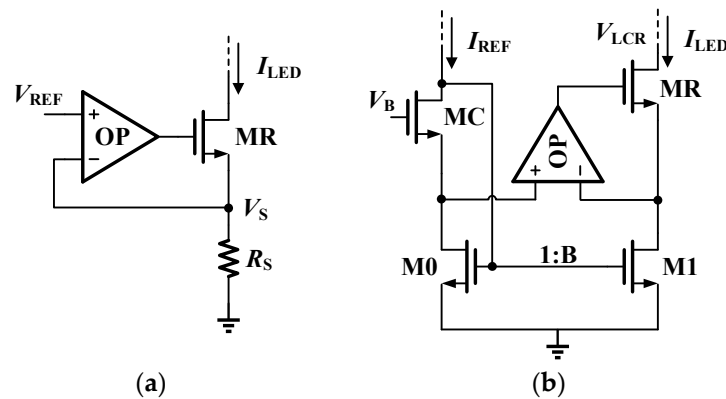


Figure 3. Typical linear current regulator circuits. (a) Source regulation structure; (b) Mirror current structure.

Figure 3b depicts another type of LCR. In this configuration, the output current I_{LED} is not directly generated as in Figure 3a, but through current mirroring. In this structure, I_{LED} is determined by $B \times I_{REF}$, where I_{REF} is a reference current and B is the current mirror ratio, which is defined by the size ratio of transistor M1 to M0. The operational amplifier (OP) plays a crucial role by ensuring that transistors M1 and M0 have the same V_{DS} , thereby guaranteeing accurate current mirroring.

Actually, M0 and M1 can work in the linear region instead of the saturation region by lowering V_B , thereby reducing V_{DS} , which helps to lower V_{LCR} and reduce excessive power [38]. The classic drain current expression for MOS transistors working in the linear region is

$$I_D = \frac{1}{2} \mu C_{OX} \left(\frac{W}{L} \right) [2(V_{GS} - V_{TH}) - V_{DS}] V_{DS} \quad (6)$$

The transistor can now be approximately equivalent to a linear resistor. As mentioned above, due to the effect of OP, the V_{DS} of M0 and M1 are kept at the same. Then the drain current of M0 and M1 has a relationship as

$$\frac{I_{D,M0}}{I_{D,M1}} = \frac{(W/L)_{M0}}{(W/L)_{M1}} = \frac{1}{B} \quad (7)$$

Clearly, accurate current mirroring is guaranteed.

Based on the circuit structure shown in Figure 3b, an improved LCR for adaptive control was designed as depicted in Figure 4. The main difference from the circuit in Figure 3b is the introduction of operational amplifier OP2. OP2, in conjunction with transistor M0, forms a negative feedback loop ensuring that the source-drain voltage V_{DS} of M0, denoted as V_A in the figure, remains equal to V_{REF} . By adjusting V_{REF} , the source-drain voltage V_{DS} of M0 can be precisely controlled. The advantage of the design is that it mitigates the impact of process variations and facilitates accurate parameter control. When V_{REF} is low, transistor M0 operates in the linear region. Transistors M1, M2, ..., MN serve as current mirror transistors corresponding to M0, each having the same size. The decoder determines whether each mirror transistor is activated or deactivated based on the binary control word CM_SW, thereby adjusting the total output current. When a mirror transistor's gate is connected to V_{COM} , it turns on; when connected to ground, it turns off. OP1 functions similarly to OP in Figure 3b, ensuring that the voltage at node D (denoted as V_D) matches V_A , thereby maintaining accurate current mirroring. The comparator COMP works as a simple circuit state detector with V_{COM} set to 2.5 V in this design. OP1 is a high-gain operational amplifier with a gain-boosting output structure [39], while OP2 is an operational amplifier with a rail-to-tail class-AB output stage [40].

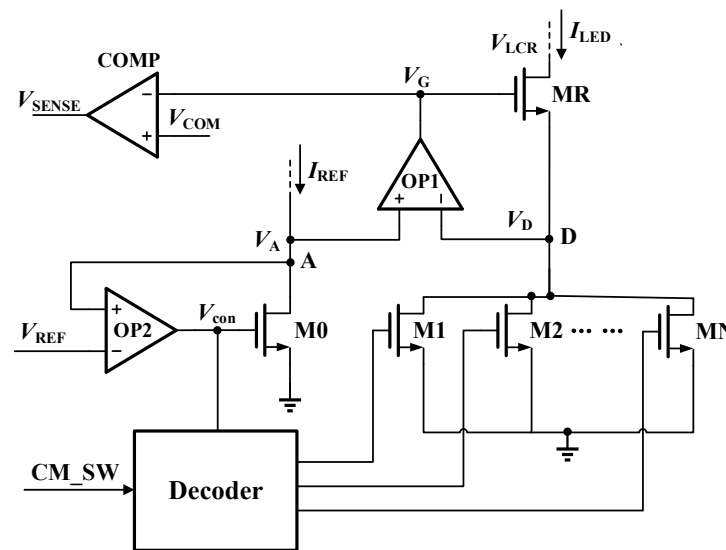


Figure 4. LCR circuit for adaptive control.

When V_{boost} is sufficiently high, even after accounting for the voltage drop across the LED chain, there remains enough V_{LCR} to ensure the proper operation of the LCR, allowing I_{LED} to reach or maintain its set value. At this condition, the gate voltage of the transistor MR is approximately 1.86 V ($V_{\text{GS,MR}} + V_{\text{REF}}$) according to the simulation results at the typical process corner, and SENSE stays at 1. If V_{boost} decreases or if I_{LED} increases and makes the forward voltage of the LED chain increase, V_{LCR} will correspondingly decrease. However, as long as transistor MR remains effective, the LCR can continue to function normally. If V_{LCR} decreases further and pushes transistor MR into deep linear region, V_G must be increased to counteract the impacts of the decreasing V_{LCR} . When V_G exceeds V_{COM} , the output of the comparator changes state, and SENSE switches to 0. This indicates that the LCR may no longer be able to maintain or achieve the set I_{LED} .

All mirror transistors have the same size, and each transistor has a size ratio of B:1 with M0. The LED current can be determined by the following equation:

$$I_{\text{LED}} = N \times B \times I_{\text{REF}} \quad (8)$$

where N is the number of activated mirror transistors and determined by CM_SW. I_{REF} is the reference current generated by the external bias circuit. It should be noted that the output current I_{LED} of the LCR actually forms the aforementioned peak current I_{peak} .

Even on the same chip, there is a certain degree of process variation, which primarily affects the device size and characteristic parameters, leading to inaccuracies or mismatches in the current mirroring. Considering the influence of the width-to-length ratio (W/L) and the threshold voltage (V_{th}), the current change caused by process variation can be expressed as follows:

$$\Delta I = \frac{\partial I}{\partial (W/L)} \Delta (W/L) + \frac{\partial I}{\partial V_{\text{th}}} \Delta V_{\text{th}} \quad (9)$$

The current mismatch is usually defined as the rate of change of current. Assuming no correlation between different influencing factors, it can be expressed as:

$$\left| \frac{\Delta I}{I} \right| = \frac{1}{I} \left(\left| \frac{\partial I}{\partial (W/L)} \Delta (W/L) \right| + \left| \frac{\partial I}{\partial V_{\text{th}}} \Delta V_{\text{th}} \right| \right) \quad (10)$$

If the transistor operates in the saturation region, according to the corresponding drain current formula, the current mismatch is

$$\left| \frac{\Delta I}{I} \right| = \left| \frac{\Delta (W/L)}{(W/L)} \right| + \left| \frac{2\Delta V_{\text{th}}}{(V_{\text{GS}} - V_{\text{th}})} \right| \quad (11)$$

While if the transistor operates in the linear region, according to the corresponding drain current formula, the current mismatch becomes

$$\left| \frac{\Delta I}{I} \right| = \left| \frac{\Delta(W/L)}{(W/L)} \right| + \left| \frac{2\Delta V_{th}}{2(V_{GS} - V_{th}) - V_{DS}} \right| \quad (12)$$

Comparing the two formulas above, it can be seen that the impact of size variation on current mismatch is the same in both the linear and the saturation regions. However, the impact of threshold voltage variation is smaller in the linear region, indicating that it is possible to achieve lower current mismatch in the linear region.

3. Simulation and Measurement Results

3.1. System and Chip Design

Based on Figure 2, this paper constructs and validates a three-channel adaptive LED driver, with each channel comprising 6 LEDs. A boost converter chip, specifically the TPS61046 from Texas Instruments, is used. The LCR is designed using a 0.18 μm complementary metal-oxide-semiconductor (CMOS) 5 V process. Since this process does not support high-voltage devices, external MP transistors are employed. Ideally, if a BCD process is available, these MP transistors could be integrated directly onto the chip. Additionally, further integration of the brightness controller is envisioned for achieving a monolithic integrated driver chip, which is a planned future work. In circuit simulation, PSPICE models are employed for the boost converter, LEDs, and MP transistors. The on-voltage parameter in the PSPICE model for LED is intentionally adjusted to introduce differences between channels. Specifically, for channel 3, the LED on-voltage is set to 3.20 V, representing a typical value. For channels 1 and 2, the parameters are adjusted to 2.73 V and 2.76 V, respectively. As described earlier, the minimum value of V_{boost} is dictated by the V_{LED} of channel 3. The brightness controller is implemented using Verilog HDL and runs at the frequency of 100 MHz. The frequency of the PWM signal is 10 KHz. The entire circuit runs in a mixed-signal simulation mode to verify the adaptive control process and assess the performance of the LCR.

In consideration of current control range and resolution, the size of M0 in the LCR is chosen to be 1 $\mu\text{m}/1 \mu\text{m}$, and its ratio to the mirror transistor is set to be 1:4. There are 240 current mirror transistors in total. The reference current I_{REF} is set to be 50 μA , resulting in a unit output current of 200 μA and a maximum output current of 48 mA. The size of MR is chosen to be 40 $\mu\text{m}/2 \mu\text{m} \times 50$. The overall layout of the LCR circuit is shown in Figure 5. All mirror transistors and their corresponding control switches are arranged in the form of a 15 $\times$ 16 array. The core area is only 325 $\mu\text{m} \times$ 310 μm , while the entire area, including pads, is 700 $\mu\text{m} \times$ 720 μm . The key transistor sizes are also summarized in Figure 5.

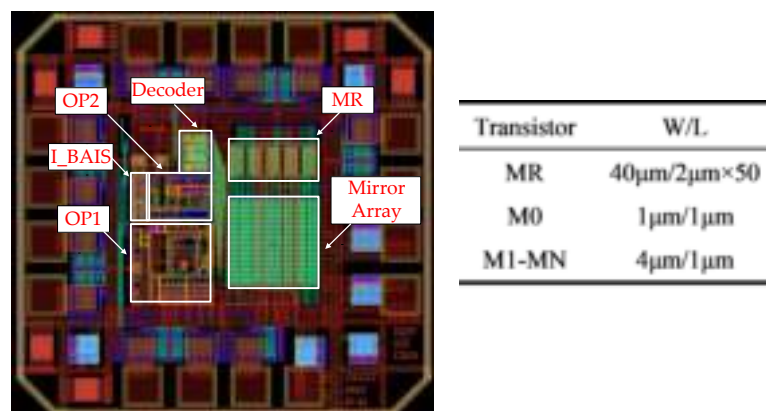


Figure 5. Layout of LCR, designed in a 0.18 μm CMOS process and the core area is 325 $\times$ 310 μm^2 . The key transistor sizes are also summarized.

3.2. Simulation Results

Figure 6 depicts the simulated output characteristics of the LCR under different process corners and temperatures. It can be seen that when the output current LCR ranges from 0 mA to 48 mA, it demonstrates good linearity under different operating conditions. Specifically, at an output current setting of 20 mA, the maximum current deviation is 13 μ A, resulting in a current error of less than 0.1%. Overall, the LCR exhibits good current linearity and accuracy within the specified current range.

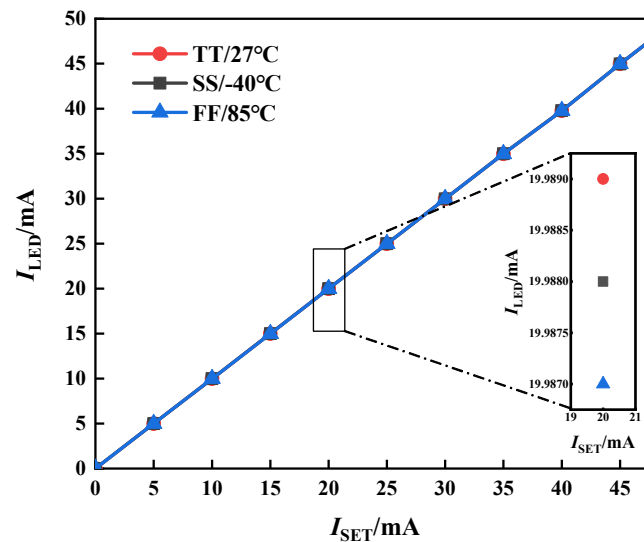


Figure 6. Output current of LCR versus current setting under various working condition.

Figure 7 depicts the voltage waveform at several essential nodes during the adaptive process. SENSE_1 , SENSE_2 and SENSE_3 represent the SENSE signals of the corresponding channels. T_0 denotes the soft start phase of the boost converter. Upon power-on, the output of the boost converter V_{boost} gradually reaches a preset value after a period of time. This preset value is the minimum voltage for the circuit to operate properly under various working conditions and is set to 25.19 V in this design. As V_{boost} increases, the SENSE signals of each channel sequentially transition to 0. T_1 marks the V_{boost} adjustment phase. During this time, the brightness controller gradually increases V_C , leading to a corresponding decrease in V_{boost} . It can be seen that when V_{boost} reaches 21.52 V, SENSE_3 transitions to 1, indicating a change in the status of channel 3. Finally, V_C stays at 0.33 V, and V_{boost} is stabilized at 21.62 V. T_2 marks the current regulation phase, during which the current of channel 1 (channel 2) increases. When SENSE_1 (SENSE_2) transitions to 1, it indicates that the status of channel 1 (channel 2) has changed. The current regulation of channel 1 (channel 2) is ceased, and the duty cycle of channel 1 (channel 2) is set according to Equation (4). T_3 represents the normal working phase, which is also called the continuous tracking phase, and SENSE signals are dynamically kept at 0.

Figure 8 illustrates the current waveform in each channel. Initially, the peak current value of each channel is set to 20 mA, with a default duty cycle of 100%. During the early stage of T_0 , V_{boost} has not been fully established, and all LEDs are in the off state, resulting in zero current in each channel. Because the LEDs in channel 3 have the largest forward voltage, the current in channel 3 is the last rising to the set value, consistent with the timing of SENSE signals in Figure 7. Following the completion of the V_{boost} adjustment phase, the current regulation phase starts. It can be seen that the peak currents in channel 1 and channel 2 are increased to 24.2 mA and 23.2 mA, respectively, with the corresponding duty cycles adjusted to 82.6% and 86.2%, ensuring uniform average currents across the three channels. When the normal working phase begins, the brightness controller will continuously monitor the state of the LCR. If the state changes, that is to say SENSE becomes 1, the peak current and the duty cycle will be updated accordingly to “pull” SENSE back

to 0. Based on the simulated current waveform, the maximum rise time and fall time are 173 ns and 74 ns, respectively, without noticeable current overshoot.

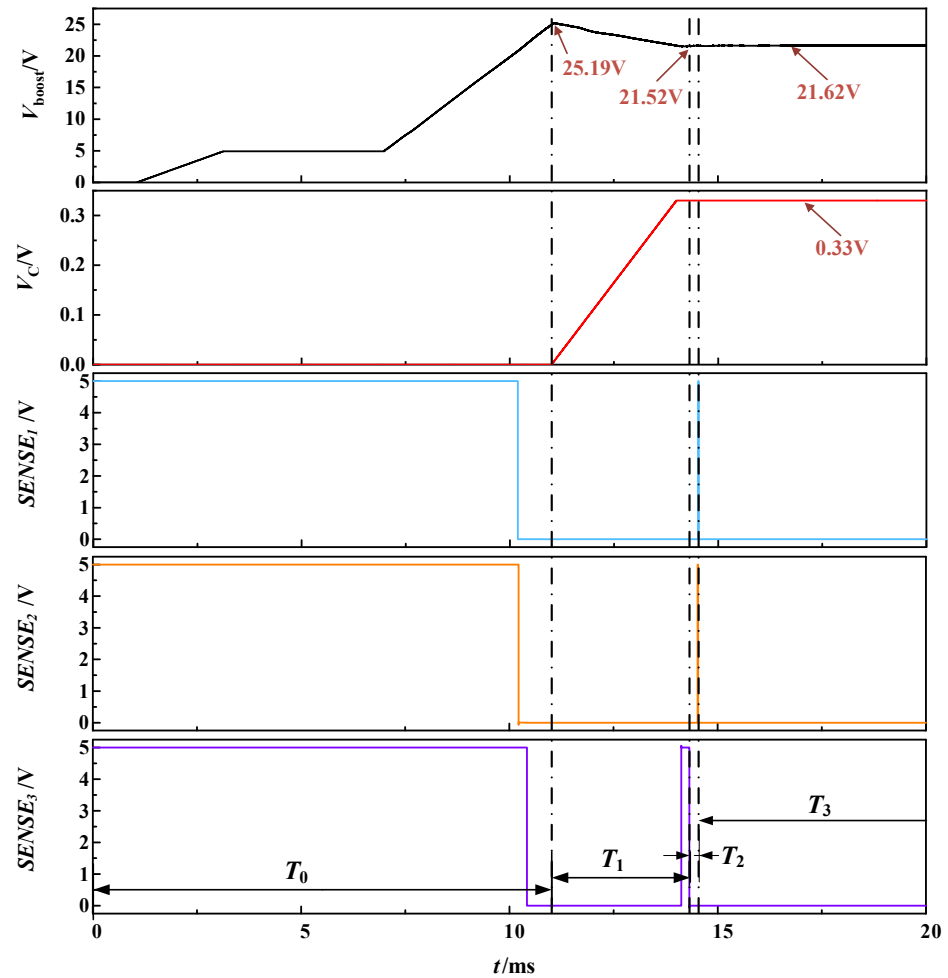


Figure 7. Voltage waveforms under the adaptive control process.

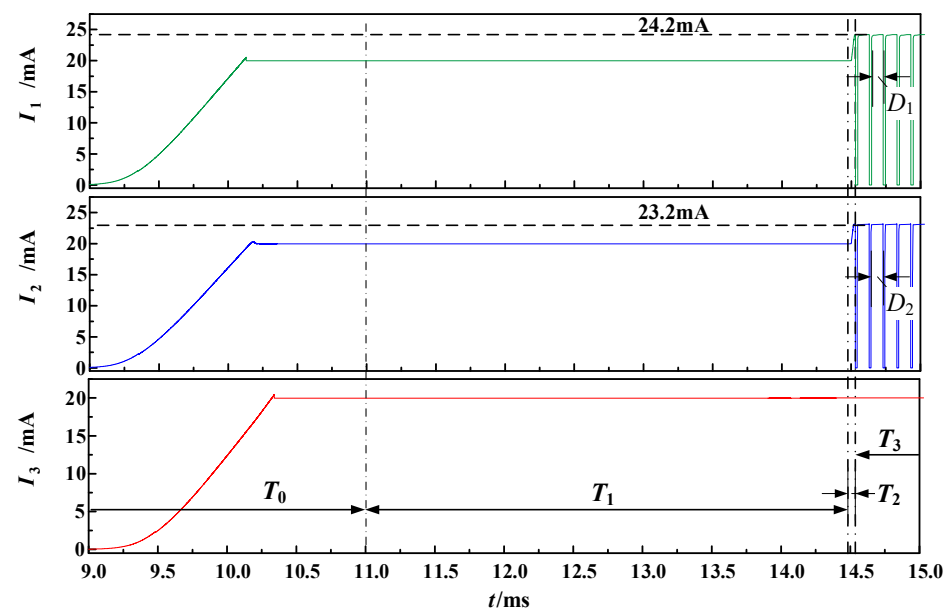


Figure 8. Current waveforms of each channel under the adaptive control process.

Figure 9 compares the power loss differences between the fixed peak current control method and the adaptive control method proposed in this paper. In the fixed peak current method, only the V_{boost} adjustment step is executed, while the peak current and the duty cycle remain unchanged. As described earlier, under the fixed peak current method, channel 3 experiences the least power loss, while channel 1 and channel 2 suffer greater power losses due to excessive voltage. The figure illustrates the power loss of the three channels under the fixed peak current method. In contrast, with the adaptive control method, both the peak current and the duty cycle are adjusted, effectively eliminating excessive voltage in channel 1 and channel 2, thereby significantly reducing power losses. As depicted in the figure, when the nominal channel current is set to 20 mA, the adaptive control method reduces power loss by 74%.

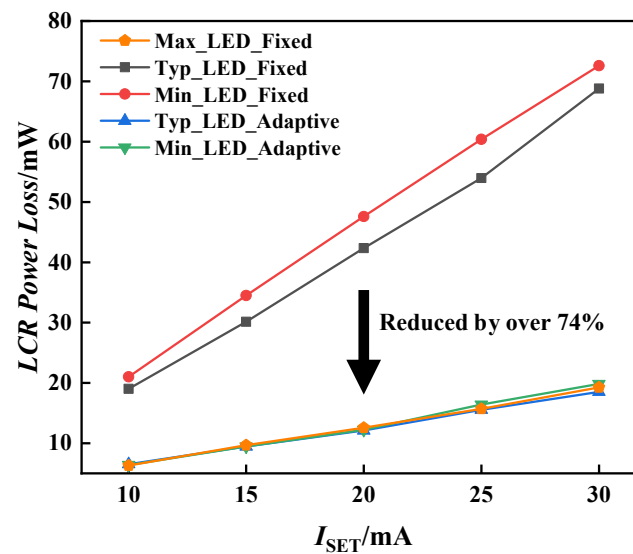


Figure 9. Power losses of LCR with the fixed peak current method and the proposed adaptive control method.

3.3. Measurement Results

Figure 10 displays a microphotograph of the fabricated chip with bonding wires, along with the chip test setup. The chip is housed in a chip-on-board (COB) package and mounted on the test board. The reference current I_{REF} is supplied by the test board. An STM32F407 MCU (from ST Microelectronics) development board serves as the brightness controller, and the integrated DAC in the MCU is used to generate V_C . A multi-channel ammeter is connected in series between each LED chain and the drain of the high-voltage MOS transistor to measure the average current of the LED channels. This ammeter supports real-time data recording via a serial port. Voltage waveforms at circuit nodes are captured using a digital oscilloscope, specifically the RTB2004 from Rohde & Schwarz. For simplicity in testing, only two LED channels are implemented, each comprising 6 LEDs. It is noteworthy that the LEDs are intentionally selected to create a difference of approximately 500 mV in the forward voltage across the LED chain between the two channels. Channel 2 has the smaller forward voltage. This deliberate difference enhances the demonstration of the circuit's effectiveness.

The indoor temperature was maintained at around 27 °C. Figure 11 illustrates the measured voltage waveforms of V_{PWM} and V_{loss} for channel 2 after the initial adaptive control process. The channel current setting I_{set} is 20 mA. The four figures correspond to the case when V_{REF} is set to 1 V, 0.6 V, 0.4 V, and 0.3 V, respectively. The duty cycle of V_{PWM} in each case is recorded as 68.57%, 68.86%, 68.43%, and 68.53%, respectively. When V_{PWM} is high, the channel is turned on, and V_{loss} in each case is measured at 1.06 V, 0.69 V, 0.49 V, and 0.36 V, respectively. When V_{PWM} transitions low, the channel is closed, and the increase in V_{loss} attributes mainly to the influence of leakage current effects. It can

be inferred from the figures that despite V_{REF} decreasing from 1 V to 0.3 V, which may cause the current mirror transistors in the LCR to enter the linear region, the accuracy of the output current can still be maintained, and the duty cycle of V_{PWM} remains consistent across these settings.

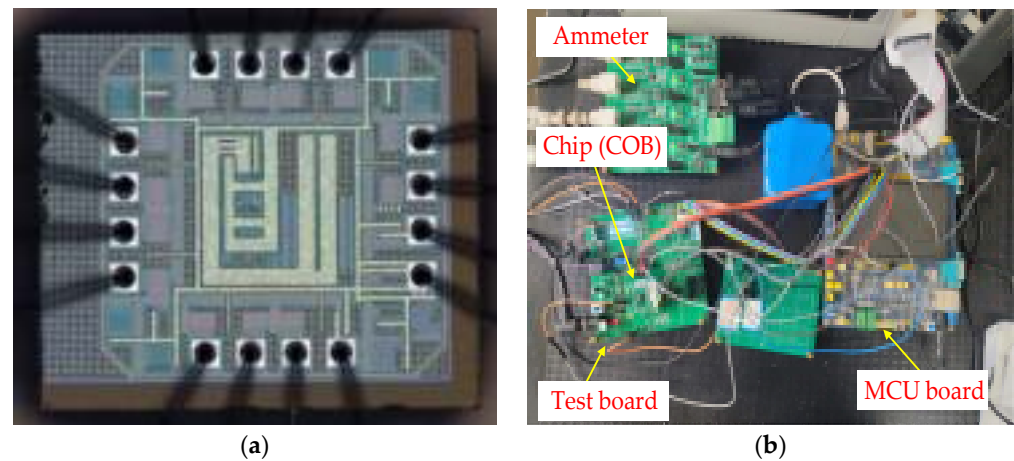


Figure 10. (a) Microphotograph of LCR chip; (b) test system.

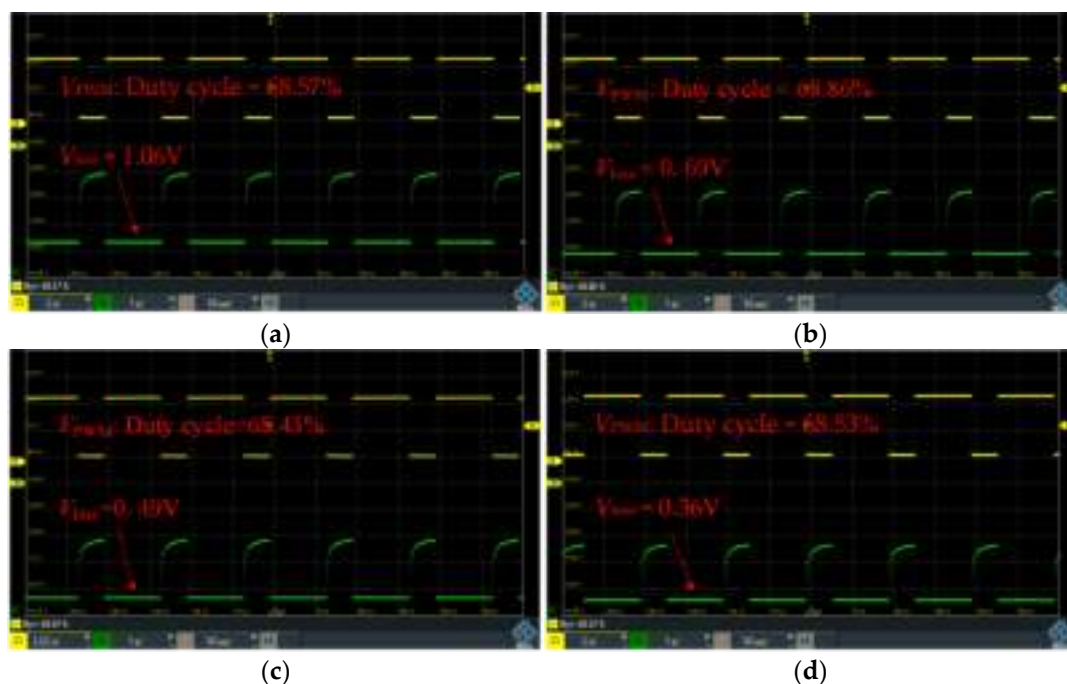


Figure 11. Voltage waveforms of V_{PWM} and V_{loss} under (a) $V_{REF} = 0.3$ V, (b) $V_{REF} = 0.4$ V, (c) $V_{REF} = 0.6$ V, and (d) $V_{REF} = 1$ V. $I_{set} = 20$ mA.

Figure 12 shows the difference in V_{loss} and power efficiency of channel 2 under both the fixed peak current control method and the adaptive control method. From Figure 12a, it is very obvious that there is an excessive voltage of about 500 mV in channel 2. With the adaptive control method, this excessive voltage is effectively mitigated through adjustments in I_{LED} , resulting in a substantial improvement in power efficiency. Specifically, when V_{REF} is set to 300 mV, the power efficiency can reach 98.1% at an I_{LED} of 20 mA. In principle, it is possible to further improve the power efficiency by reducing V_{REF} even further, with the upper limit constrained by V_{COM} , which is up to the power supply.

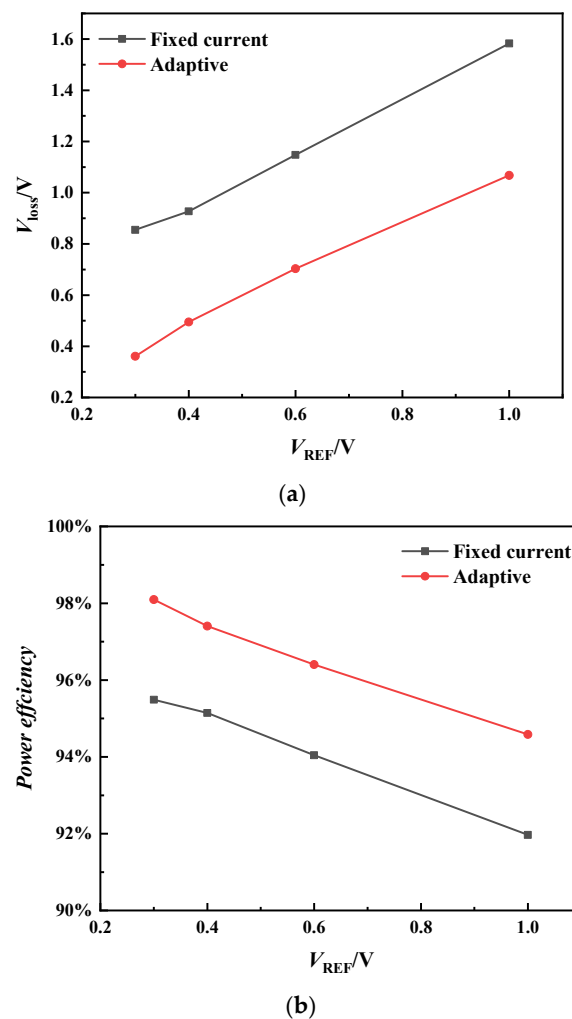


Figure 12. Comparison between the fixed peak current method and the proposed adaptive control method in regard to (a) V_{loss} and (b) power efficiency.

Table 1 compares various parameters and performance metrics between the proposed adaptive driver and other existing references. The proposed design demonstrates advantages such as a wide output current range and high power efficiency. In fact, Table 1 only reveals the efficiency of a single channel. However, more importantly, the proposed adaptive method aims to improve the efficiency of each channel in a multi-channel LED driver. It effectively compensates for excessive voltage on the LCR through adaptive control with moderate implementation complexity.

Table 1. Performance parameters comparison.

Ref	[17]	[41]	[42]	[43]	This work
Process	0.35 μm BCD	0.5 μm CMOS	40 nm CMOS	40 nm CMOS	0.18 μm CMOS
Core area (mm^2)	N/A ⁺	N/A ⁺	N/A ⁺	N/A ⁺	0.1
Power supply (V)	6–27	3–4.2	3.2–4.2	3.2–4.2	5
Maximum I_{LED} (mA)	30	20	25	25	48
LED chain config.	8	2	6	6	6
V_{boost} (V)	26.41	5.36	22.5	20	18.96 [*]
V_{loss} (mV)	~400	~300	~675	~480	360 [*]
Power Efficiency	98.4%	94.4%	97%	97.6%	98.1% [*]
V_{boost} adjustment	Yes	Yes	Yes	Yes	Yes
Channel-by-channel adjustment	No	No	No	No	Yes

^{*} Measured at $I_{set} = 20$ mA, $\delta_{set} = 100\%$. ⁺ Not available.

4. Summary

In this paper, an adaptive high-efficiency LED backlight driver is designed to address the issue of excessive power loss caused by variations in the forward voltage of LEDs. By adjusting the peak current, the consistency in the forward voltage of the LED chain within each channel is achieved. This minimizes the voltage on the LCR in each channel, thereby optimizing the excessive power loss in each channel and the total power efficiency. Furthermore, adjusting the duty cycle ensures brightness consistency across each channel. Based on the 0.18 μm 1P4M CMOS process, an LCR circuit suitable for adaptive control was designed and fabricated. The circuit utilizes a current mirror structure and MOS transistors operating in the linear region to ensure current accuracy and power efficiency. The designed LCR circuit supports an output current range of 0–48 mA. To verify the effectiveness of the adaptive control circuit, an experimental two-channel LED backlight driver was constructed using the designed LCR chips. Each channel consists of 6 LEDs, and the forward voltage difference between channels is intentionally expanded to approximately 500 mV. The test results demonstrate that the presented adaptive control circuit eliminates this forward voltage difference and reduces the excessive power loss. When the reference voltage is set to 300 mV, a power efficiency of 98.1% is achieved at 20 mA. In fact, the experimental two-channel driver can be extended to multi-channel implementation. While only an LCR chip is designed and fabricated in this study, the MP transistors and the brightness controller can both be integrated to realize a monolithic integrated driver chip using BCD processes.

Author Contributions: Conceptualization, W.Z. and Q.Z.; methodology, X.X., Q.Z., C.J. and Z.Z.; software, X.X.; validation, X.X., Q.Z., C.J. and Z.Z.; formal analysis, X.X.; investigation, X.X., Q.Z., C.J. and Z.Z.; resources, X.X. and Q.Z.; data curation, X.X. and Q.Z.; writing—original draft preparation, X.X. and Q.Z.; writing—review and editing, W.Z. and X.X.; supervision, W.Z. and J.T.; project administration, W.Z. and J.T.; funding acquisition, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Hunan Provincial Department of Education Scientific Research Project (23A0260) and Changsha University of Science and Technology Graduate Practice Innovation Project (CLSJCX24079).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tsao, J.Y. Solid-state lighting: Lamps, chips, and materials for tomorrow. *IEEE Circuits Devices Mag.* **2004**, *20*, 28–37. [[CrossRef](#)]
2. Yadlapalli, R.T.; Narasipuram, R.P.; Kotapati, A. An overview of energy efficient solid state LED driver topologies. *Int. J. Energy Res.* **2020**, *44*, 612–630. [[CrossRef](#)]
3. Morrow, R.C. LED lighting in horticulture. *HortScience* **2008**, *43*, 1947–1950. [[CrossRef](#)]
4. Cho, J.; Park, J.H.; Kim, J.K.; Schubert, E.F. White light-emitting diodes: History, progress, and future. *Laser Photonics Rev.* **2017**, *11*, 1600147. [[CrossRef](#)]
5. Surkanti, P.R.; Mehrotra, D.; Furth, P.M. Multi-channel LED driver with accurate current matching for portable applications. In Proceedings of the 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), Dallas, TX, USA, 4–7 August 2019; pp. 941–944.
6. Pal, N.; Fish, A.; McIntyre, W.; Griesert, N.; Winter, G.; Eichhorn, T.; Pilawa-Podgurski, R.; Hanumolu, P.K. A 91% efficient 30 V hybrid boost-SC converter based backlight LED driver in 180 nm CMOS. In Proceedings of the 2020 IEEE Custom Integrated Circuits Conference (CICC), Boston, MA, USA, 22–25 March 2020; pp. 1–4.
7. Pressman, A. *Switching Power Supply Design*; McGraw-Hill, Inc.: New York, NY, USA, 1997.
8. Wang, Y.; Alonso, J.M.; Ruan, X. A review of LED drivers and related technologies. *IEEE Trans. Ind. Electron.* **2017**, *64*, 5754–5765. [[CrossRef](#)]
9. Gupta, N.; Verma, D. A review and analysis of topologies of light emitting diode (LED) drivers. In Proceedings of the 2021 IEEE 2nd International Conference on Electrical Power and Energy Systems (ICEPES), Bhopal, India, 10–11 December 2021; pp. 1–6.
10. Oh, T.-J.; Cho, A.; Ki, S.-L.; Hwang, I.-C. A low-power and low-cost digitally-controlled boost LED driver IC for backlights. In Proceedings of the 2012 IEEE Asian Solid State Circuits Conference (A-SSCC), Kobe, Japan, 12–14 November 2012; pp. 237–240.

11. Lee, S.-W.; Choe, H.-J.; Yun, J.-J. Performance improvement of a boost LED driver with high voltage gain for edge-lit LED backlights. *IEEE Trans. Circuits Syst. II Express Briefs* **2017**, *65*, 481–485. [\[CrossRef\]](#)
12. Chen, H.; Zhang, Y.; Ma, D. A SIMO parallel-string driver IC for dimmable LED backlighting with local bus voltage optimization and single time-shared regulation loop. *IEEE Trans. Power Electron.* **2011**, *27*, 452–462. [\[CrossRef\]](#)
13. Kim, H.-C.; Yoon, C.S.; Jeong, D.-K.; Kim, J. A single-inductor, multiple-channel current-balancing LED driver for display backlight applications. *IEEE Trans. Ind. Appl.* **2014**, *50*, 4077–4081. [\[CrossRef\]](#)
14. Chen, J.-H.; Wang, S.-C.; Liu, Y.-H.; Cheng, Y.-S.; Yang, Z.-Z. RGB LED backlight driving system with dynamic voltage regulation capability. In Proceedings of the 2013 IEEE 10th International Conference on Power Electronics and Drive Systems (PEDS), Kitakyushu, Japan, 22–25 April 2013; pp. 349–353.
15. Liu, P.-J.; Hong, Z.-Y.; Hung, C.-C. A single-stage low-power AC–DC RGB-LED driver with switching capacitor control scheme. *IEEE Trans. Ind. Electron.* **2020**, *68*, 3103–3112. [\[CrossRef\]](#)
16. Kim, J.-S.; Lee, Y.-K.; Lee, J.-S.; Shin, Y.-K.; Tark, J.-H.; Ryu, K.-C.; Yang, B.-D. A fast-switching current-pulse driver for LED backlight. In Proceedings of the 2009 IEEE International Symposium on Circuits and Systems, Taipei, Taiwan, 24–27 May 2009; pp. 1775–1778.
17. Hong, S.-I.; Han, J.-W.; Kim, D.-H.; Kwon, O.-K. A double-loop control LED backlight driver IC for medium-sized LCDs. In Proceedings of the 2010 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 7–11 February 2010; pp. 116–117.
18. Huang, Y.-C.; Chen, H.-W. A novel fast-switching current-pulse driver for LED backlight applications. In Proceedings of the 2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Nantou, Taiwan, 27–29 May 2016; pp. 1–2.
19. Ahn, H.-A.; Hong, S.-K.; Kwon, O.-K. A fast switching current regulator using slewing time reduction method for high dimming ratio of LED backlight drivers. *IEEE Trans. Circuits Syst. II Express Briefs* **2016**, *63*, 1014–1018. [\[CrossRef\]](#)
20. Han, H.; Zhang, F.; Liu, M. PWM dimming method for capacitor-clamped current-sharing circuit in LED backlight system. *IEEE J. Emerg. Sel. Top. Power Electron.* **2018**, *6*, 1190–1197. [\[CrossRef\]](#)
21. Hong, S.-I.; Nam, K.-S.; Jung, Y.-H.; Ahn, H.-A.; In, H.-J.; Kwon, O.-K. Highly power-efficient and reliable light-emitting diode backlight driver IC for the uniform current driving of medium-sized liquid crystal displays. *J. Inf. Disp.* **2012**, *13*, 73–82. [\[CrossRef\]](#)
22. Lim, B.-M.; Ko, Y.-H.; Jang, Y.-S.; Kwon, O.-H.; Han, S.-K.; Lee, S.-G. A 200-V 98.16%-efficiency buck LED driver using integrated current control to improve current accuracy for large-scale single-string LED backlighting applications. *IEEE Trans. Power Electron.* **2015**, *31*, 6416–6427. [\[CrossRef\]](#)
23. Parthasarathy, M.K.; Ganesan, N. A novel single-inductor eight-channel light-emitting diode driver for low power display backlight applications. *Turk. J. Electr. Eng. Comput. Sci.* **2018**, *26*, 2130–2139. [\[CrossRef\]](#)
24. He, J.-P.; Zhang, B.; Gao, X.; Wu, C.-D.; Li, Z.-J.; Zhou, J.-Y. A high accuracy weak voltage LED analog dimming method. *IEEE Access* **2019**, *7*, 172362–172373. [\[CrossRef\]](#)
25. Lin, C.-L.; Chen, S.-C.; Deng, M.-Y.; Ho, Y.-H.; Lin, C.-A.; Tsai, C.-L.; Liao, W.-S.; Liu, C.-I.; Wu, C.-E.; Peng, J.-T. AM PWM driving circuit for mini-LED backlight in liquid crystal displays. *IEEE J. Electron Devices Soc.* **2021**, *9*, 365–372. [\[CrossRef\]](#)
26. Schubert, E.F. *Light-Emitting Diodes* (2018); Rensselaer Polytechnic Institute: Troy, NY, USA, 2018.
27. Liu, C.-H.; Hsieh, C.-Y.; Hsieh, Y.-C.; Tai, T.-J.; Chen, K.-H. SAR-controlled adaptive off-time technique without sensing resistor for achieving high efficiency and accuracy LED lighting system. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2009**, *57*, 1384–1394.
28. Lohaus, L.; Rossius, A.; Dietrich, S.; Wunderlich, R.; Heinen, S. A dimmable LED driver using resistive DAC feedback control for adaptive voltage regulation. In Proceedings of the 2013 IEEE Energy Conversion Congress and Exposition, Denver, CO, USA, 15–19 September 2013; pp. 3126–3133.
29. Hsieh, C.-Y.; Wei, Y.-S.; Chen, K.-H.; Lin, T.-C. Efficient LED driver with an adaptive reference tracking technique. In Proceedings of the 2009 16th IEEE International Conference on Electronics, Circuits and Systems (ICECS 2009), Yasmine Hammamet, Tunisia, 13–16 December 2009; pp. 291–294.
30. Liu, P.-J.; Hsu, S.-R.; Chang, C.-W.; Liao, C.-Y.; Chien, L.-H. Dimmable white LED driver with adaptive voltage feedback control. In Proceedings of the 2015 IEEE 2nd International Future Energy Electronics Conference (IFEEEC), Taipei, Taiwan, 1–4 November 2015; pp. 1–4.
31. Jung, Y.H.; Hong, S.K.; Kwon, O.K. A fast transient LED driver with adaptive frequency control according to load variation for large-sized LCD backlights. *J. Soc. Inf. Disp.* **2017**, *25*, 712–724. [\[CrossRef\]](#)
32. Hsu, Y.-C.; Lin, J.-Y.; Chen, C.-C.-P. Area-saving and high-efficiency RGB LED driver with adaptive driving voltage and energy-saving technique. *Energies* **2018**, *11*, 1422. [\[CrossRef\]](#)
33. Yong, W.; Baoxing, D.; Yi, L.; Ting, L.; Yintang, Y. A Multi-channel High Efficiency Boost LED Backlight Driver with High Dimming Ratio. In Proceedings of the 2017 International Conference on Electronic Industry and Automation (EIA 2017), Suzhou, China, 23–25 June 2017; pp. 110–116.
34. Hsieh, Y.-T.; Liu, B.-D.; Wu, J.-F.; Fang, C.-L.; Tsai, H.-H.; Juang, Y.-Z. A high-dimming-ratio LED driver for LCD backlights. *IEEE Trans. Power Electron.* **2012**, *27*, 4562–4570. [\[CrossRef\]](#)
35. Chiu, H.-J.; Lo, Y.-K.; Chen, J.-T.; Cheng, S.-J.; Lin, C.-Y.; Mou, S.-C. A high-efficiency dimmable LED driver for low-power lighting applications. *IEEE Trans. Ind. Electron.* **2009**, *57*, 735–743. [\[CrossRef\]](#)
36. Hu, Y.; Jiang, L.; Wen, W.; Yan, W.; Ding, R. A new adaptive drive voltage approach for LED driver. In Proceedings of the 2019 IEEE Applied Power Electronics Conference and Exposition (APEC), Anaheim, CA, USA, 17–21 March 2019; pp. 721–726.

37. Wu, C.-H.; Chen, C.-L. High-efficiency current-regulated charge pump for a white LED driver. *IEEE Trans. Circuits Syst. II Express Briefs* **2009**, *56*, 763–767.
38. Mu, H.; Geng, L.; Liu, J. A high precision constant current source applied in LED driver. In Proceedings of the 2011 Symposium on Photonics and Optoelectronics (SOPO), Wuhan, China, 16–18 May 2011; pp. 1–4.
39. Sansen, W.M. *Analog Design Essentials*; Springer Science & Business Media: Berlin, Germany, 2007; Volume 859.
40. Hogervorst, R.; Tero, J.P.; Eschauzier, R.G.; Huijsing, J.H. A compact power-efficient 3 V CMOS rail-to-rail input/output operational amplifier for VLSI cell libraries. *IEEE J. Solid-State Circuits* **1994**, *29*, 1505–1513. [[CrossRef](#)]
41. Surkanti, P.R.; Mehrotra, D.; Gangineni, M.; Furth, P.M. Multi-string LED driver with accurate current matching and dynamic cancellation of forward voltage mismatch. In Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 6–9 August 2017; pp. 229–232.
42. Shahroury, F.R.; Ahmad, H.H.; Abuishmais, I. Design Aspects of a Single-Output Multi-String WLED Driver Using 40 nm CMOS Technology. *J. Low Power Electron. Appl.* **2022**, *12*, 5. [[CrossRef](#)]
43. Abuishmais, I.; Shahroury, F.R.; Ahmad, H. A Design Methodology of High-Efficiency Dimmable Current Sink for Current-Regulated Drivers. *Electronics* **2022**, *11*, 2566. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Object Detection and Monocular Stable Distance Estimation for Road Environments: A Fusion Architecture Using YOLO-RedeCa and Abnormal Jumping Change Filter

Hejun Lv , Yu Du *, Yan Ma  and Ying Yuan

Beijing Key Laboratory of Information Service Engineering, College of Robotics, Beijing Union University, Beijing 100101, China; lyuhejun@163.com (H.L.); jqrmayan@bnu.edu.cn (Y.M.); yueeing@163.com (Y.Y.)

* Correspondence: duyue@bnu.edu.cn

Abstract: Enabling rapid and accurate comprehensive environmental perception for vehicles poses a major challenge. Object detection and monocular distance estimation are the two main technologies, though they are often used separately. Thus, it is necessary to strengthen and optimize the interaction between them. Vehicle motion or object occlusions can cause sudden variations in the positions or sizes of detection boxes within temporal data, leading to fluctuations in distance estimates. So, we propose a method to integrate a detector based on YOLOv5-RedeCa, a Bot-Sort tracker and an anomaly jumping change filter. This combination allows for more accurate detection and tracking of objects. The anomaly jump filter smooths distance variations caused by sudden changes in detection box sizes. Our method increases accuracy while reducing computational demands, showing outstanding performance on several datasets. Notably, on the KITTI dataset, the standard deviation of the continuous ranging results remains consistently low, especially in scenarios with multiple object occlusions or disappearances. These results validate our method's effectiveness and precision in managing dual tasks.

Keywords: automatic driving technique; object detection; monocular distance measurement; Kalman filter



Citation: Lv, H.; Du, Y.; Ma, Y.; Yuan, Y. Object Detection and Monocular Stable Distance Estimation for Road Environments: A Fusion Architecture Using YOLO-RedeCa and Abnormal Jumping Change Filter. *Electronics* **2024**, *13*, 3058. <https://doi.org/10.3390/electronics13153058>

Academic Editor: Felipe Jiménez

Received: 24 June 2024

Revised: 19 July 2024

Accepted: 26 July 2024

Published: 2 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Efficient perception and understanding of the road environment are crucial for autonomous driving technology [1]. However, achieving comprehensive perception of the surrounding environment for vehicles causes a challenging problem. This comprehensive perception system performs multiple tasks. It detects various objects around the vehicle. Additionally, it obtains the distances between these objects and the vehicle. Therefore, we focus on methods that achieve dual tasks. Our methods simultaneously detect objects around the vehicle and determine the distances to these objects. More details can be found in Figure 1.

In autonomous driving, accurate object detection is crucial for dynamic distance estimation, directly affecting the system's safety and efficiency [2]. Dynamic distance estimation, as part of higher-level perception, can simulate the intuitive perception of human drivers.

Existing distance measurement methods based on radar and stereo vision, although effective, face challenges such as high costs, system complexity, and poor real-time performance. Furthermore, it is worth exploring whether current bird's-eye view (BEV) perception technology requires deeper support from human visual characteristics [3]. Existing monocular distance measurement methods lack robustness when dealing with sudden value changes, which directly affects the perception and decision-making quality of autonomous driving systems in dynamic environments.

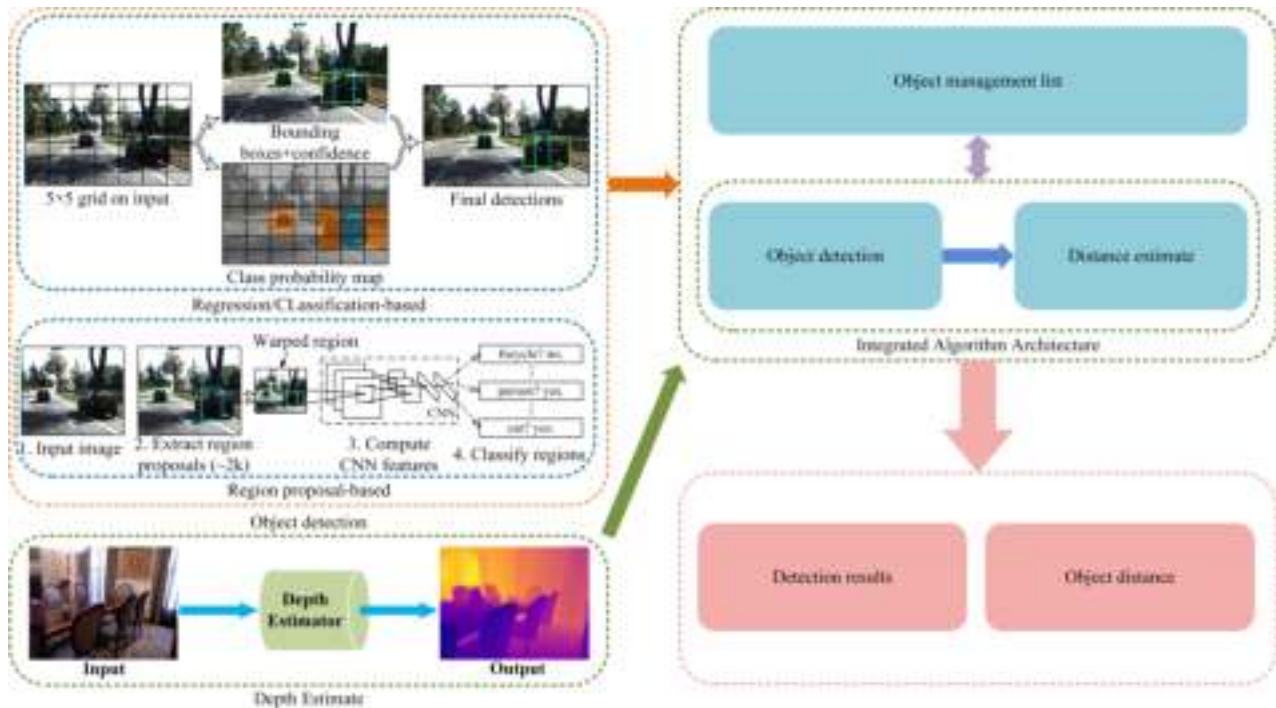


Figure 1. The left side is the main framework of object detection with monocular depth estimation. On the right is the architecture of the algorithm we are going to implement and the output.

The goal of this study is to propose a monocular distance measurement method. This method aims to both detect objects and determine the distance between the objects and the vehicle. By combining the YOLOv5-RedeCa and anomaly jump filter methods, the approach intends to enhance the accuracy and robustness of dynamic distance estimation, thereby addressing perception and decision-making challenges in dynamic environments. Our method demonstrates significant potential in feature extraction and distance estimation. Additionally, it will provide more reliable and richer perceptual information for autonomous driving systems, thereby significantly improving driving safety and efficiency. This makes our method a promising candidate for breakthrough advancements in autonomous driving technology.

Reviewing previous work in the field of 2D object detection, region proposal-based methods (e.g., R-CNN) [4] have significantly improved detection accuracy through a two-stage detection approach, performing exceptionally well in complex backgrounds. However, they are relatively slow in computation. In contrast, although single-stage detectors (e.g., YOLO) [5] slightly compromise on accuracy, they directly predict the positions and categories of objects in an image through regression/classification, simplifying the model structure and computation process, making them suitable for real-time processing [6]. Therefore, we use them as the baseline for our detector.

Object detection alone cannot fully perceive the environment without distance information [7]. Traditional depth estimation methods use monocular cues or manually designed features [8,9], but they often underperform in varying depth scenarios.

Deep learning, especially with convolutional neural networks (CNNs), offers direct pixel-to-depth mapping [10]. Multi-scale networks and scale-invariant loss functions have enhanced depth estimation accuracy. Unlike traditional methods, deep learning automatically generates disparity maps and depth information without extensive annotation, typically using video sequences or stereo images. However, these methods require substantial computational resources.

In autonomous driving, continuous object detection information from the time domain is used for tracking objects and distance estimation. However, vibrations caused by vehicle motion and occlusions can disrupt the stability of objects in the video sequence. This can

lead to sudden changes in the position or size of the detection boxes, as shown in Figure 2. Therefore, it is necessary to smooth the detection boxes.

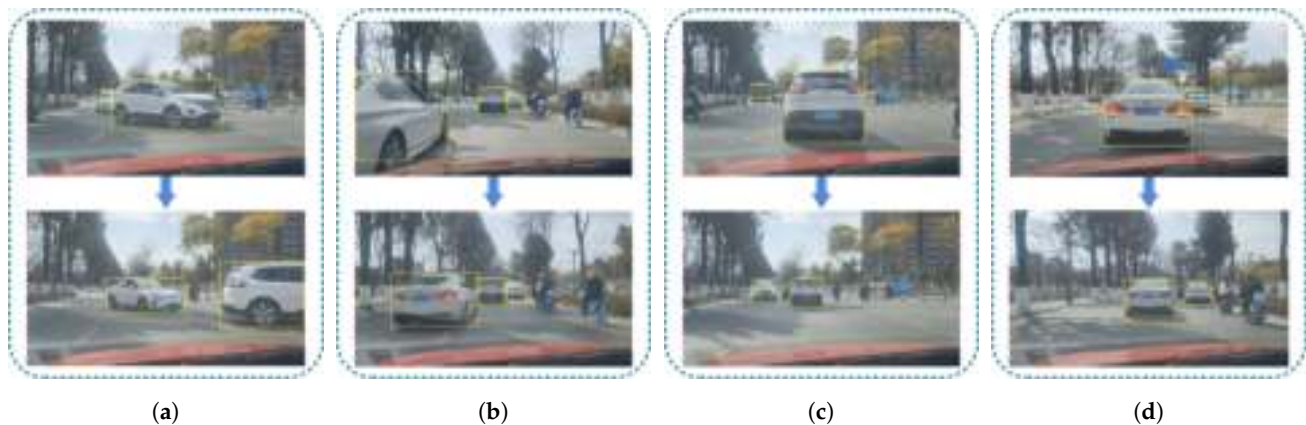


Figure 2. The bounding boxes for non-ideal and ideal captures of road scene objects are illustrated. (a–d) represent consecutive frames from four distinct video sequences. In (a,b), the bounding box dimensions exhibit abrupt changes when the target object undergoes sudden appearance, disappearance, or occlusion events. Conversely, (c,d) depict scenarios where the target is clearly visible, and the bounding box accurately encapsulates the target object under normal conditions.

Though algorithms for 2D object detection have made significant progress, they remain specialized. Similarly, monocular depth estimation algorithms have advanced considerably. However, research in these two fields remains relatively separate. Integration between object detection and depth estimation is still lacking. In the existing research, the object detection often lacks integrated depth estimation, and the depth estimation does not include object recognition capabilities. This situation highlights an urgent need. Developing an integrated system that combines object detection and depth estimation is essential. Such a system would optimize the interaction between the two techniques. Consequently, it would improve their efficiency in practical applications.

Therefore, we propose and design a holistic algorithm architecture that fuses object detection and distance estimation. Our main contributions are as follows:

1. We design the RedeCa module to enhance feature extraction capability, enabling more precise object detection in road traffic scenarios. To ensure real-time performance, we use a deep separable convolutional network architecture.
2. We integrate a multi-frame association mechanism, to improve the algorithm's robustness in cases of object occlusion or temporary loss, enabling simultaneous tracking and ranging of multiple different objects.
3. We design an anomaly jumping change filter to enhance distance estimation accuracy and robustness. This filter eliminates anomalies caused by sudden changes in detection box size within a frame. It smooths the variation in object distance. Consequently, the reliability of ranging is improved.

Our proposed algorithm won the gold medal in The 3rd China (shenyang) Intelligent Connected Vehicles Conference. It also achieved the bronze medal in the perception & decision control group at the 2023 World Intelligent Driving Challenge Metaverse Virtual Simulation Competition.

2. Related Work

In the field of autonomous driving, how to make the vehicle fully perceive its surrounding environment is a challenging problem. This comprehensive perception includes not only the detection of various surrounding objects, but also the acquisition of distances between these objects and the vehicle. Currently, the use of visible light cameras is a common and cost-effective approach to addressing this problem. Equipped with photosensitive elements, visible light cameras can capture color images of the real world. Researchers

conduct in-depth analysis of these color images to accomplish key perception tasks such as object detection and distance estimation.

Traffic Object Detection. Object detection is one of the core problems in the field of computer vision [11]. It aims to classify and determine the position of objects in images. Deep convolutional neural networks (DCNNs) have provided strong support for the development of object detection techniques with their outstanding feature representation capabilities [12,13]. Commonly, the main architectures for object detection can be divided into two categories: region proposal-based object detection methods and regression/classification-based object detection methods [6]. Each of them has its own characteristics, addressing different technical challenges.

Representative models of region proposal-based object detection architectures include RCNN [4] and its subsequent improvements such as SPPNet [14], Fast RCNN [15], Faster RCNN [16], Mask RCNN [17], and R-FCN [18]. The first stage of these architectures generates candidate region proposals. The second stage performs detailed classification and bounding box refinement on these regions. This staged processing greatly enhances detection accuracy, particularly exhibiting excellent performance in complex backgrounds. For instance, to address the issue of slow proposal generation speed, Faster RCNN introduces the region proposal network (RPN). It is significant to improve the quality and speed of proposals. Mask RCNN further predicts object masks by adding a parallel branch, enabling the model to perform both object detection and instance segmentation simultaneously. This is particularly valuable in applications requiring fine-grained visual information.

To simplify model structure and computational processes, regression/classification-based object detection architectures are designed to predict object positions and categories directly on images, such as OverFeat [19], the YOLO series [20–25], SSD [26], DSSD [27], and FSSD [28]. The main advantage of these architectures is their high detection speed, making them suitable for real-time processing scenarios. Among them, the YOLO series stands out. By reframing the detection problem as a single regression task, it can simultaneously predict the categories and positions of multiple objects in a single network forward pass. To effectively address detection challenges for objects of various sizes and improve the model's ability to detect small objects, SSD introduces the concept of multi-scale feature maps.

Monocular Depth Estimation. The pursuit of determining the distance between objects and the camera has remained a pivotal research focus within the domain of computer vision since its inception. In the initial approaches, monocular depth estimation primarily relied on Shape from Shading (SFS) and Depth from Focus/Defocus (DFF/DFD). SFS infers the object's shape based on changes in illumination on its surface [29]. DFF/DFD techniques estimate depth using focused and defocused information in images along with camera calibration parameters [30]. The fundamental assumption is that the focused parts of the image are clearer, while the defocused parts gradually become blurry. Although effective in specific scenarios, these methods often rely on additional scene assumptions and exhibit subpar performance when confronted with scenes featuring significant depth of field.

With the advancement of machine learning technology, researchers have begun to frame the single-pixel depth estimation problem as a probabilistic learning problem based on random fields. For instance, optimizing depth maps using Markov Random Field (MRF) [8,31], and modeling posterior probabilities with Conditional Random Field (CRF) for depth estimation [9]. The enhanced computational power of Graphics Processing Units (GPUs) and the availability of large-scale image datasets have driven deep learning, paving the way for new research directions in depth estimation.

Deep learning techniques typically require real depth data as training labels. These techniques directly map pixels to depth values by constructing convolutional neural networks (CNNs). Refs. [32,33] were the first to apply CNNs in the field of depth estimation. They employed a multi-scale network strategy. A global coarse estimation network provides initial distance estimates. Additionally, a local refinement network optimizes the details. They introduced a scale-invariant loss function to handle errors. Subsequent studies

have further improved estimation accuracy by refining network structures [29,34–36] and integrating auxiliary information [37,38]. Some research does not rely on a large amount of labeled depth data. Instead, they utilize video sequences or stereo images to automatically derive disparity maps and depth information using deep learning algorithms. For example, Ref. [39] process the left view using a convolutional neural network. This generates a disparity map. Then, they reconstruct the left view by combining it with the right view through inverse mapping. The reconstruction error is used as a supervisory signal for training. Ref. [40] simultaneously processes left and right views to optimize disparity map generation, enabling accurate depth estimation even when only monocular images are used during testing. Additionally, Ref. [41] derives pose and depth from video sequences of consecutive frames, demonstrating the effectiveness of combining multiple network models for view reconstruction. While these methods achieve good performance, they come at a significant computational cost.

Significant progress has been made in 2D object detection and monocular depth estimation. But the two remain largely separate in research, with no effective integration observed. At present, object detection systems often lack the capability for range estimation. And the depth estimation system does not integrate the object recognition ability. This separation suggests a need for development. A holistic algorithmic architecture is necessary. Such an architecture should integrate object detection and range estimation. This integration would enhance the synergy between these two techniques. Consequently, it would optimize their performance in practical applications.

3. Method

To address the integration of object detection and distance estimation, we design and develop an algorithmic architecture. It comprises three main components: a detector based on YOLOv5-RedeCa, a multi-object tracker, and an anomaly jumping change filter. In autonomous driving scenarios, vehicles often encounter complex and dynamic environments with multiple autonomously moving objects that may suddenly appear, become occluded, or overlap [21]. These situations impose stringent requirements on the accuracy and robustness of the algorithm. Additionally, due to the high speed of vehicle movement, the algorithm must process sensor data and react within milliseconds to ensure vehicle safety. Therefore, autonomous driving systems require a perception algorithm with high accuracy and robustness.

For these purposes, we designed the RedeCa module, which enhances the architecture of the object detection algorithm, improving the precision of object localization in road scenarios. Furthermore, we integrate the Bot-Sort tracker. This tracker employs a multi-frame association mechanism. It enhances the algorithm's robustness when encountering object occlusion or temporary loss. Thus, the method can simultaneously track multiple objects and estimate their distances. The approach also periodically evaluates tracking continuity, monitors significant changes in object state, and checks the stability of the tracking process. To enhance the accuracy and robustness of distance estimation, we also integrate an anomaly jumping change filter. The filter effectively smooths the variation in object distance and eliminates potential anomalies caused by sudden changes in detection box size. Thereby, it can improve the reliability of distance measurement. The overall architecture is illustrated in Figure 3.

By implementing YOLOv5-RedeCa, our method enhances accuracy in object recognition and refines bounding box precision across video frames. Furthermore, we compute the distances between objects and the camera using a sophisticated physical imaging model.

To effectively track and measure multiple objects, we integrate the Bot-Sort tracker. This multi-frame association mechanism significantly enhances robustness in scenarios with occlusions or temporary loss of objects. Even if the object temporarily vanishes from view in certain frames, it continues to track and measure distance consistently. We assign unique identifiers to newly detected objects using an incremental sequence. Then, it utilizes the Bot-Sort to match these with existing objects, distinguishing between those that have

newly appeared and those already being tracked. For objects that are successfully matched, their status information is updated, and original identifiers are preserved. Conversely, new objects are allocated fresh tracking records and identifiers. If an object is not detected in the current frame, its entry is temporarily preserved. If the object fails to reappear within a predetermined period, it will be removed from the tracking list.

Now, most monocular ranging methods primarily focus on the accuracy of individual images, overlooking the correlation of distance changes across consecutive frames in the temporal domain. Therefore, we propose a multi-frame data analysis strategy, which integrates Kalman filtering. It not only enhances the precision and robustness of distance estimation, but also assists in the detailed analysis of the object's motion patterns and trends in distance changes. Kalman filtering smooths distance variations by analyzing changes in the same object across consecutive frames. It also eliminates potential errors caused by abrupt changes in the size of single-frame detection boxes.

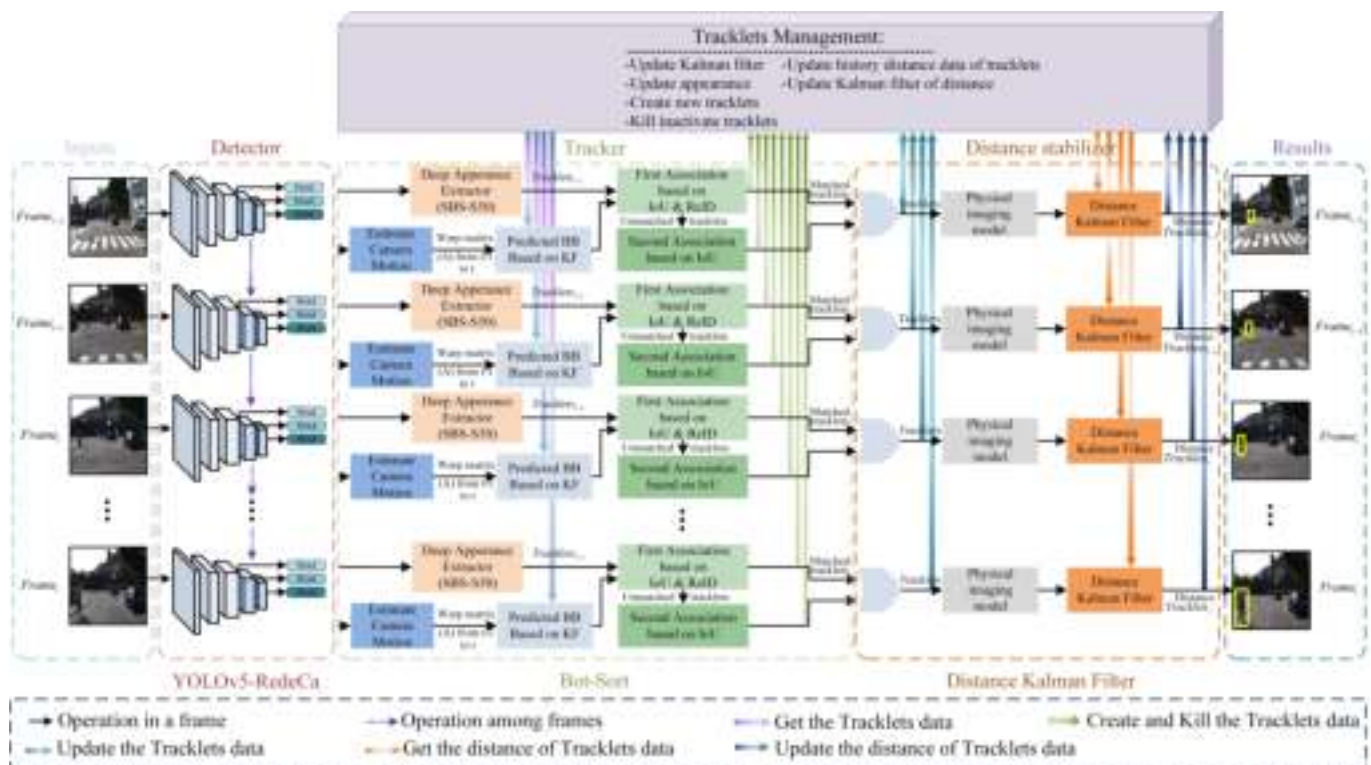


Figure 3. The Graph of the overall architecture.

3.1. Object Detection

To ensure accurate object recognition and precise bounding box extraction in each video frame, we employed the YOLOv5 object detection algorithm. It serves as the core framework for our detection module, which is based on deep convolutional neural networks. YOLOv5 predicts the class and position of objects, i.e., the bounding boxes, directly from the input image in a single forward pass. Compared to the two-stage detection methods, which first extract candidate regions and then classify them, YOLOv5 considerably improves detection speed and efficiency. It uses a deep convolutional network to extract features from images, progressively drawing out increasingly complex characteristics [42]. It enhances its ability to recognize various objects. By integrating a Feature Pyramid Network (FPN), YOLOv5 combines feature maps from different levels, thereby improving its detection capabilities across a range of object sizes.

Considering the diversity of object sizes in road scenes, the accuracy of object detection, and the specific requirements of road traffic, we have improved the YOLOv5 network architecture. First, we integrated the RedeCa module into the residual blocks. This allows for more accurate detection of smaller objects under varying lighting conditions and

weather changes. Then, we replaced the traditional convolutions in the residual blocks with depthwise separable convolutions. This change reduces the number of parameters in the network and enhances the real-time performance. The architecture of YOLOv5-RedeCa network is shown in Figure 4.

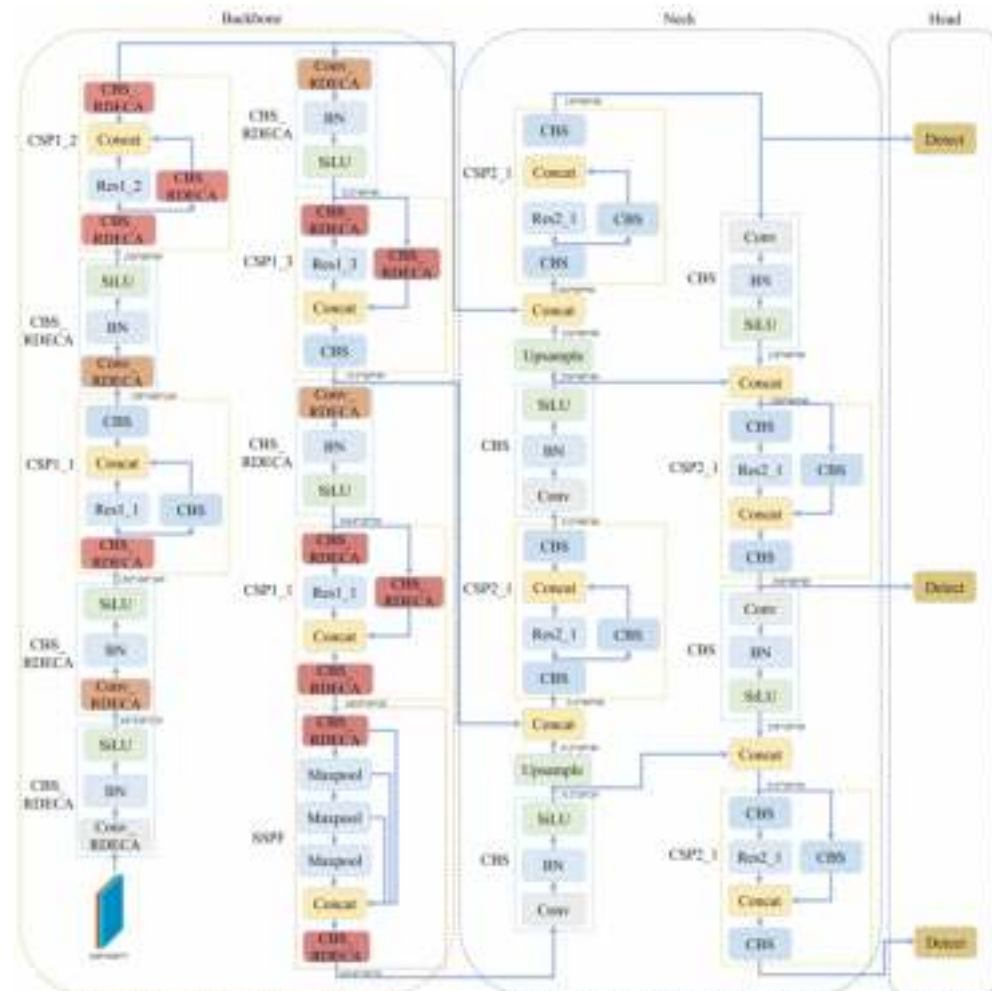


Figure 4. The YOLOv5-RedeCa Network Architecture Diagram.

RedeCa Module

YOLOv5 has demonstrated outstanding performance across a wide range of detection scenarios. However, the complex road environment demands a versatile approach to accommodate the diverse types and sizes of traffic objects. To address this challenge, we optimized YOLOv5's network structure, specifically by designing the RedeCa module to enhance recognition accuracy for traffic objects of various sizes. As shown in Figure 5, the RedeCa module is based on a residual block. It incorporates a sequence of residual depthwise separable convolution blocks and ECA modules to enhance feature extraction capabilities.

Depthwise separable convolutions markedly reduce the number of parameters and computational load in models. This is achieved by breaking down standard convolutions into two simpler operations: depthwise convolutions and pointwise convolutions. The process is detailed in Figure 6. It involves first applying depthwise convolution independently to each input channel. Then, pointwise convolution aggregates these outputs using 1×1 convolution kernels. This approach greatly reduces the parameter count and computational demands, lowering memory usage and energy consumption. Consequently, it makes the model more suitable for deployment on devices with limited resources.

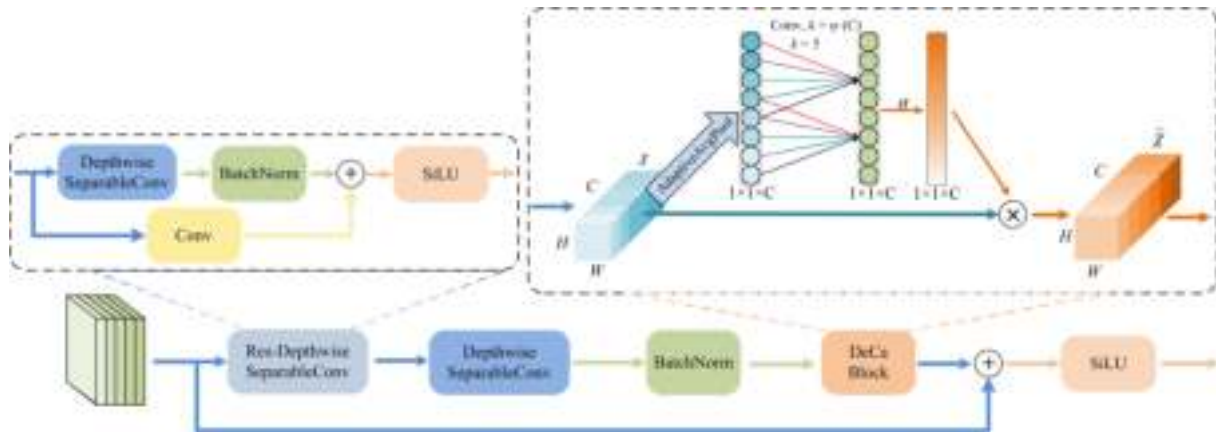


Figure 5. The Graph of RedeCa Structure.

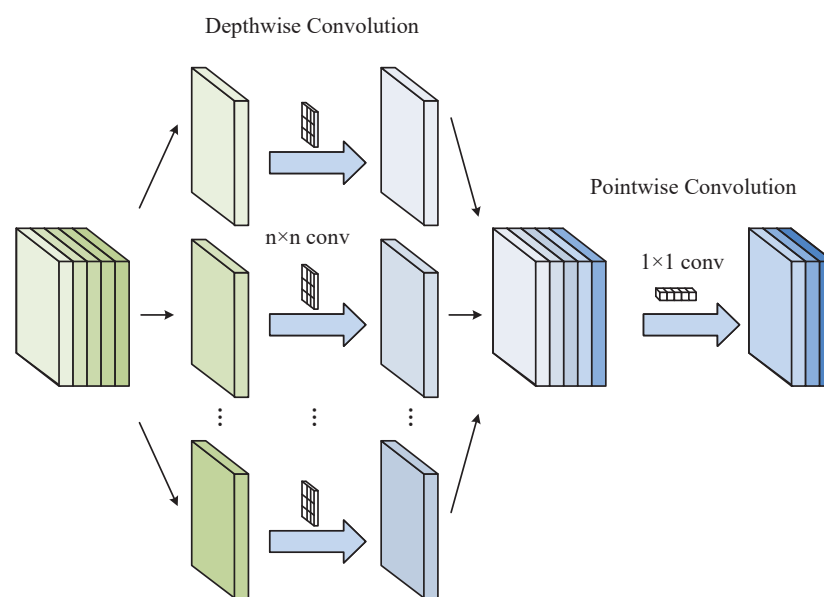


Figure 6. The Graph of Depthwise Separable Convolutional Structure.

Depthwise separable convolutions minimize the risk of overfitting, which is particularly crucial when training models on limited data.

By reducing the number of parameters, training becomes simpler and the likelihood of overfitting is reduced. It enhances the model's generalization capability in real-world applications. The computational complexity associated with depthwise separable convolutions is detailed below.

Depthwise Convolution:

$$Param_{DC} = K \times K \times C_{in} \quad (1)$$

$$FLOPS_{DC} = H_{out} \times W_{out} \times K \times K \times C_{in} \quad (2)$$

Pointwise Convolution:

$$Param_{PC} = C_{in} \times C_{out} \quad (3)$$

$$FLOPS_{PC} = H_{out} \times W_{out} \times C_{in} \times C_{out} \quad (4)$$

Depthwise Separable Convolutions:

$$Param_{DW} = Param_{DC} + Param_{PC} \quad (5)$$

$$FLOPS_{DW} = FLOPS_{DC} + FLOPS_{PC} \quad (6)$$

where $Param_{DC}$ quantifies the parameter count associated with depthwise convolution, and $FLOPS_{DC}$ delineates the computational demands requisite for depthwise convolution. Correspondingly, $Param_{PC}$ specifies the parameter count for pointwise convolution, whereas $FLOPS_{PC}$ characterizes its computational demands. Additionally, $Param_{DW}$ is indicative of the parameter count in depthwise separable convolution, with $FLOPS_{DW}$ elucidating the computational demands for this type of convolution.

The RedeCa module is initiated with a depthwise separable convolution block that integrates principles of a residual network. It uses identity mapping to improve the learning efficiency of deep network layers. It combines the high parameter efficiency of depthwise separable convolutions with the rapid learning capability of residual connections, thereby enhancing the module's feature extraction ability. And then, a standard depthwise separable convolution block is integrated to effectively reduce the model's complexity and computational demands. Finally, the deca module adjusts the dependencies between channels, dynamically optimizing feature responses to improve the accuracy of object detection.

The mathematical formulation of the RedeCa module is as follows:

$$F_{DWConv}^{Res} = SiLU(BN(DWConv(F^*)) + Conv(F^*)) \quad (7)$$

$$F_{DWConv} = BN(Pointwise(Depthwise(F_{DWConv}^{Res}))) \quad (8)$$

$$F_{RedeCa} = SiLU(F_{DWConv} \times \sigma(Conv(AdaptAvgPool(F_{DWConv}))) + F^*) \quad (9)$$

where F^* represents the feature map input from the previous layer.

3.2. Enhancing the Robustness of Multi-Frame Association in Distance Estimation

In object detection and distance estimation in road traffic scenes, it is not enough to operate only on a single object. In practical applications, it is necessary to accurately estimate the distance of multiple objects simultaneously. And these objects may undergo sudden appearances, occlusions, overlaps, or disappearances. Our proposed method integrates the Bot-Sort tracker, assigning a unique identifier to each object in every frame, facilitating effective object identification and tracking. Although existing monocular ranging algorithms have improved accuracy on individual images, they often overlook the coherence of distance changes for the same object across consecutive frames. Ignoring this coherence can lead to fluctuations in distance measurements, especially when the bounding box size of the object detection changes drastically over a short period. Our method enhances distance estimation accuracy and robustness by integrating Kalman filtering techniques over time series data. The approach smooths out variations in object distances and eliminates anomalies caused by sudden changes in single-frame detection box sizes, resulting in superior performance in real-time road traffic scenarios.

3.2.1. Bot-Sort

In the domain of visual analysis for road traffic, the mere detection and distance estimation of individual objects are not adequate. Paramount to this field is the capability to accurately measure the distances of multiple objects within a scene and allocate distinct identifiers to each for effective tracking and differentiation. Our method integrates the Bot-Sort tracker, an advanced algorithm that adeptly identifies and analyzes every object within video frames, ascertaining new objects for tracking. The methodology of the Algorithm 1 is delineated in the form of pseudocode [43].

Algorithm 1 Frame-based object Tracking System Workflow

```

1: Input: Video frames
2: Output: Tracked object trajectories
3: for each frame in Video do
4:   Detect potential objects using YOLOX
5:   Estimate camera motion transformation matrix from previous frame
6:   Extract features for detected objects using SBS-S50 model
7:   Predict object locations in the current frame using Kalman Filter
8:   First association based on fused IoU and ReID scores
9:   Match current detections with existing tracklets
10:  Second association for unmatched tracklets based on IoU
11:  Update tracklets' state information
12:  Create new tracklets and assign new unique identifiers
13:  Remove inactive tracklets if not detected within a certain period
14:  (Optional) Interpolate missing parts in tracklets
15: end for
16: return Completed trajectories for all objects

```

BoT-SORT stands as a superior multi-object tracking system that significantly improves tracking accuracy through the synergistic integration of motion and visual appearance data. The system leverages advanced Kalman filters alongside camera motion compensation techniques to adeptly handle real-world dynamics. It distinctively amalgamates Intersection over Union (IoU) with cosine distance measures from Re-identification (ReID) algorithms to establish consistent and reliable object associations, particularly in densely dynamic environments.

3.2.2. Object Identification and Maintenance

Our method incorporates the following logic for object identification and maintenance. Firstly, for newly detected objects, the method assigns each object a unique identifier through an incremental sequence to differentiate between different tracking objects. Subsequently, the method applies the Bot-Sort algorithm. It compares newly discovered objects with those already tracked to determine their status.

For successfully matched objects, the system updates their status information in the tracking list, while keeping their original unique identifiers unchanged. Newly appearing objects are created as new tracking entries and assigned new unique identifiers. If a tracked object is not detected in the current frame, possibly due to occlusion or leaving the field of view, the method temporarily retains its entry and ID. If the object does not reappear within the set time, it will be removed from the tracking list.

3.2.3. Anomaly Jumping Change Filter

In modeling object motion on the image plane, we typically utilize a discrete Kalman filter equipped with a constant velocity model. The state vector of this filter is defined as a septuple, $x = [x_c, y_c, s, a, \dot{x}_c, \dot{y}_c, \dot{s}]^T$, where (x_c, y_c) specifies the 2D coordinates of the object's center within the image plane. Here, s signifies the scale (area) of the bounding box, and a indicates its aspect ratio. To augment the precision and resilience of distance estimation, to stabilize the variations in object distance, and to address anomalies potentially triggered by abrupt changes in detection box sizes within a frame, the state vector is expanded to an octuple, $x = [x_c, y_c, a, h, \dot{x}_c, \dot{y}_c, \dot{a}, \dot{h}]^T$. Accordingly, the state vector of the Kalman Filter (KF) is defined as depicted in Equation (10), and the measurement vector is delineated in Equation (11).

$$x_k = [x_c(k), y_c(k), a(k), h(k), \dot{x}_c(k), \dot{y}_c(k), \dot{a}(k), \dot{h}(k)]^T \quad (10)$$

$$z_k = [z_{x_c}(k), z_{y_c}(k), z_a(k), z_h(k)]^T \quad (11)$$

This approach, which integrates Kalman filtering into time series analysis, is designed to significantly improve the accuracy and robustness of distance estimation. The Kalman filter algorithm methodically smoothens fluctuations in object distances by conducting an analysis of historical data. It effectively addresses anomalies that arise from sudden changes in the size of detection boxes within individual frames. The algorithm's state prediction is formulated in Equations (12) and (13), while the observation update mechanism is detailed in Equations (14)–(16).

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k \quad (12)$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \quad (13)$$

where $\hat{x}_{k|k-1}$ represents the predicted state, F_k is the state transition matrix, B_k denotes the control input model, u_k is the control vector, $P_{k|k-1}$ is the predicted covariance, and Q_k signifies the process noise covariance.

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \quad (14)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - H_k \hat{x}_{k|k-1}) \quad (15)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (16)$$

where K_k denotes the Kalman gain, H_k represents the observation matrix, z_k is the actual observation, R_k indicates the observation noise covariance, and I is the identity matrix.

4. Experiments

4.1. Datasets

COCO Datasets. The Common Objects in Context (COCO) dataset, developed by the Microsoft team, is a vital asset in computer vision research. It contains over 330,000 images that are richly annotated across 80 object categories. By providing a diverse, large-scale, and challenging benchmark, the COCO dataset has facilitated extensive research activities [44]. Consequently, the method we proposed has been rigorously evaluated on the COCO dataset to compare its performance with leading algorithms in the field. Table 1 summarizes the experimental setup and dataset partitioning, illustrating our method within the COCO dataset.

Table 1. COCO Dataset Training, Validation, and Test Set Splits.

Dataset Split	Number of Images	Covered Categories
Training Set	118,287	80 classes
Validation Set	5000	80 classes
Test Set	40,670	80 classes

VOC Datasets. The PASCAL Visual Object Classes (VOC) dataset serves as a crucial benchmark in the field of computer vision, supporting tasks such as object detection. It encompasses annotations for 20 different object categories, including class labels and bounding boxes. The PASCAL VOC challenge series played a pivotal role in advancing vision-based technologies [45]. Additionally, we evaluate the improved YOLOv5 on the VOC dataset and compare it with other popular methods. Table 2 provides a detailed description of partitioning for the VOC dataset.

Table 2. VOC Dataset Training, and Validation Set Splits.

Dataset Split	Number of Images	Covered Categories
Training Set	11,540	20 classes
Validation Set	10,991	20 classes

KITTI Datasets. The KITTI dataset is a crucial benchmark in the field of autonomous driving research. It encompasses a variety of real-world driving scenarios including urban, rural, and highway environments. It provides comprehensive annotations for tasks such as 3D object detection, tracking, and depth estimation [46]. We use the KITTI dataset to validate the accuracy of our proposed method in distance estimation, and to demonstrate its applicability and effectiveness in realistic driving conditions.

4.2. Implementation Details

4.2.1. Training Setup

In order to ensure the fairness and consistency of the model training effect, the details of the hardware, software and experimental environment are shown in Table 3. The training parameter Settings are shown in Table 4.

Table 3. Experiment environment.

Environment	Version
Operating System	Ubuntu 20.04
CPU	AMD Ryzen 9 5900HX
GPU	Nvidia GeForce RTX 3080 16 G
Compiling Environment	Python 3.9.0
CUDA	12.2
Deep Learning Framework	Pytorch 1.13.1
Compiler	Pycharm 2022.3.1

Table 4. Training parameters.

Parameter	Value
Batch Size	16
Init Learning Rate	0.1
Final Learning Rate	0.0001
Image Size	640 × 640
Optimizer	SGD
Workers	2

4.2.2. Loss Function

The loss function used in the YOLOv5-RedeCa model is calculated as follows:

$$L(t_p, t_{gt}) = \sum_{k=0}^K [\alpha_k^{balance} \alpha_{box} \sum_{i=0}^{S^2} \sum_{j=0}^B \prod_{kij} L_{CIoU} + \alpha_{obj} \sum_{i=0}^{S^2} \sum_{j=0}^B \prod_{kij} L_{obj} + \alpha_{cls} \sum_{i=0}^{S^2} \sum_{j=0}^B \prod_{kij} L_{cls}] \quad (17)$$

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \quad (18)$$

where K , S^2 , and B represent the output feature map, cell, and the number of anchors on each cell, respectively. α_x represents the weight of the corresponding item. α_{box} , α_{obj} , and α_{cls} take values of 0.05, 0.7, and 0.3, respectively. We based the parameter values of α_{box} , α_{obj} , and α_{cls} on the original YOLOv5 algorithm. Using a stable and recognized baseline configuration is essential for consistent performance evaluation. $\alpha_k^{balance}$ is used to balance the weights of the output feature maps at each scale, with a default value of [4.0, 1.0, 0.4].

$\mathbb{I}_{kij}^{obj}$ indicates whether the k th output feature map, the i th cell, and the j th anchor box are positive samples. If it is a positive sample, it is 1; otherwise, it is 0. t_p , and t_{gt} are the predicted vector and ground-truth vector.

The loss function of CIOU is as follows:

$$L_{CIOU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \quad (19)$$

where b represents the center points of the predicted box, and b^{gt} represents the ground-truth box. α is the balance factor between the loss caused by aspect ratio and the loss caused by IoU. v is the normalization of the difference in aspect ratio between the predicted box and the ground-truth box. They are mathematically defined as follows:

$$\rho(b, b^{gt}) = \|b - b^{gt}\|_2 \quad (20)$$

$$\alpha = \frac{v}{(1 - IoU) + v} \quad (21)$$

$$v = \frac{4(\arctan \frac{w_G}{h_G} - \arctan \frac{w_P}{h_P})^2}{\pi^2} \quad (22)$$

The loss functions for object and class are defined as follows:

$$L_{obj}(p_o, p_{IoU}) = BCE_{obj}^{sig}(p_o, p_{IoU}; w_{obj}) \quad (23)$$

$$L_{cls}(c_p, c_{gt}) = BCE_{cls}^{sig}(c_p, c_{gt}; w_{cls}) \quad (24)$$

where (w_G, h_G) represents the width and height of the object box and (w_P, h_P) represents the width and height of the predicted box. p_o is the confidence score of the object in the predicted box. p_{IoU} is the confidence score of the object in the predicted box. BCE_{cls}^{sig} and BCE_{cls}^{sig} represent binary cross-entropy loss. w_{obj} and BCE_{cls} represent the weights of positive samples.

4.3. Evaluation Index

In the following experiments, we evaluate the accuracy of object detection against the accuracy and stability of distance estimation using the following metrics.

- **Precision:** This metric is pivotal for assessing the accuracy of object detection models. It quantifies the ratio of instances accurately identified as objects to all cases labeled as objects by the model. The formula is defined as

$$Precision = \frac{TP}{TP + FP} \quad (25)$$

where TP (True Positives) represents the count of correctly detected objects, and FP (False Positives) denotes the instances erroneously classified as objects.

- **Recall:** Recall evaluates the proportion of true objects that the model successfully detects out of the entire set of actual objects. A higher recall implies that the model is capable of identifying most true objects, albeit potentially increasing the rate of false positives. The formula for recall is

$$Recall = \frac{TP}{TP + FN} \quad (26)$$

with FN (False Negatives) indicating the number of objects that were not detected.

- **Average Precision (AP):** AP calculates the mean value of precision across varying levels of recall. It can be determined for each category, and the Mean Average Precision (MAP) represents the average of APs across all categories. MAP is extensively used

to gauge the comprehensive performance in tasks involving detection of multiple object classes.

$$mAP = \frac{1}{classes} \sum_{i=1}^{classes} \int_0^1 P(R) d(R) \quad (27)$$

- **Standard Deviation (SD):** SD serves as a statistical measure to quantify the extent of variation or dispersion from the mean within a dataset. In the domain of time-domain analysis, it is pivotal to precisely evaluate the closeness of distance estimations derived from various monocular ranging algorithms to the true values, as well as for gauging their dispersion. The calculation of standard deviation is defined by the following formula:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (f(x_i) - \mu_i)^2} \quad (28)$$

where σ denotes SD, $f(x_i)$ represents the estimated distance at time instance i , and μ_i corresponds to the actual value at that instance. This metric is crucial for assessing the reliability and accuracy of these algorithms in real-world applications.

4.4. Comparison Experiment

We have developed an improved YOLOv5 method, specifically optimized for detecting small objects in road scenes under various lighting and weather conditions. To evaluate the performance of our method in terms of feature representation and real-time processing capabilities, we conducted a series of experiments on the COCO and VOC datasets. Our improved methods, denoted according to scale as Ours-s, Ours-m, and Ours-l, were compared with several state-of-the-art object detection methods.

We used Precision, mAP, and Recall as the primary performance metrics. Additionally, we considered complexity by evaluating the number of parameters (Params) and floating-point operations per second (FLOPS). We performed a comparative analysis of different methods on the COCO and VOC datasets, ensuring the robustness of the experimental design and the validity of the data analysis. The results are presented in the accompanying Table 5.

Table 5. Detection results of different algorithms on COCO and VOC datasets.

Method	Precision (%)	COCO		Precision (%)	VOC		Params (M)	FLOPS (G)
		mAP@.5 (%)	Recall (%)		mAP@.5 (%)	Recall (%)		
SSD	50.7	45.1	38.8	31.4	54.7	57.3	26.3	65.7
Faster R-CNN	53.1	48.0	46.0	34.1	57.5	54.7	60.4	93.6
Mask-RCNN	51.9	48.0	45.7	33.9	57.2	69.1	63.4	147.0
YOLOv5s	63.5	54.3	52.1	69.4	60.3	52.9	7.1	16.1
YOLOv5m	70.4	61.3	55.4	82.5	74.3	67.2	21.0	48.5
YOLOv5l	72.9	64.0	59.7	82.5	75.4	69.3	46.2	108.6
YOLOv6s	64.7	52.8	46.2	65.3	58.3	51.6	18.5	45.3
YOLOv7s	65.8	47.6	49.5	66.8	58.3	52.5	6.1	13.2
YOLOv8s	68.1	56.3	53.1	68.8	56.5	53.0	11.2	28.8
YOLOv9s	67.1	55.7	50.7	66.7	61.7	54.1	9.6	38.8
YOLOv10s	66.3	57.2	51.8	68.4	60.7	56.1	8.1	24.9
Ours-s	67.2	55.0	51.3	67.0	57.5	53.5	6.7	13.2
Ours-m	71.6	61.5	57.9	81.9	77.5	73.4	17.2	34.6
Ours-l	73.6	64.4	60.1	82.5	80.0	76.0	34.3	70.4

Experimental results on the COCO dataset show that our optimized Ours-m model achieved 71.6% Precision, 61.5% mAP, and 57.9% Recall. These results are the best among all methods using similar computational resources. On the VOC dataset, the model also excelled, registering 77.5% mAP@.5 and 73.4% Recall. Notably, despite having fewer parameters than the equivalent scale YOLOv5 model, Ours-s, Ours-m and Ours-l surpassed

other methods in Precision and mAP. It demonstrates our success in improving performance while optimizing computational efficiency.

Compared to SSD and Faster R-CNN, Ours-m demonstrated significant advantages on both the COCO and VOC datasets. It was particularly effective in Recall for small object detection. For instance, on the COCO dataset, the Recall of Ours-l was 21.3% higher than that of SSD and 14.1% higher than Faster R-CNN, highlighting the effectiveness of the RedeCa module in enhancing recall capabilities.

Although Mask-RCNN is designed for instance segmentation, the Ours-l model excelled in object detection tasks. On the VOC dataset, it achieved an mAP@.5 of 80.0%, compared to Mask-RCNN's 57.2%. This emphasizes that our improved model delivers outstanding detection performance, even in domains specialized by Mask-RCNN.

We also controlled for model complexity while striving for higher performance. Compared to the original YOLOv5, the Ours series demonstrated superior small object detection in complex environments with fewer parameters. On the COCO dataset, Ours-l improved mAP and Recall by 0.4% each over YOLOv5l. Our methods surpass existing methods in key performance metrics while maintaining computational efficiency, laying a solid foundation for practical applications.

We compared our method with the latest object detection methods from YOLOv6 to YOLOv10 [47–51]. Our method matches YOLOv8 and YOLOv9 in terms of accuracy, and surpasses YOLOv6, v8, v9, and v10 in model lightweighting and computational load. The data presented in Table 5 showed that while our model achieves accuracy comparable to the latest YOLO versions, it still has benefits in model lightweighting and computational load. These advantages contribute to the deployment and operational efficiency of our method in practical applications.

The RedeCa module enhances our method's ability of representing features, especially in accurately handling edge and detail information, which is crucial for small object detection. It also increases the adaptability to varying lighting and weather conditions, thereby boosting robustness in complex environments.

In road traffic scenarios, real-time performance is crucial. We evaluated our method against popular methods in terms of FPS and parameter count, as detailed in Table 5. Our method's parameter count and computational complexity are markedly lower than those of Faster R-CNN and Mask-RCNN. Compared to the baseline YOLOv5, our method achieves comparable performance while reducing both parameter count and computational complexity.

The experimental results show that by adding the RedeCa module to YOLOv5, we improve small object detection accuracy markedly while maintaining high recall rates. These findings support strongly further exploration of deep learning methods for small object detection in complex environments, confirming the effectiveness and potential of our enhancements.

To validate the superior accuracy and robustness of our method, we conducted a comprehensive quantitative and qualitative comparative analysis. We compared our method with popular monocular distance measurement methods in road traffic scenarios. The results are shown in Figure 7.

As shown in Figure 7, our monocular distance measurement method effectively reduces potential errors caused by sudden changes in the detection bounding box size. This leads to reduced dispersion in distance measurements, ensuring they remain more consistent with actual distances over time.

We also conducted a qualitative analysis to compare our method with mainstream methods. Unlike methods that generate depth maps for the entire image to obtain object distances, our method detects objects and measures their distances simultaneously, improving efficiency. The distance measurement results are shown in Figure 8.

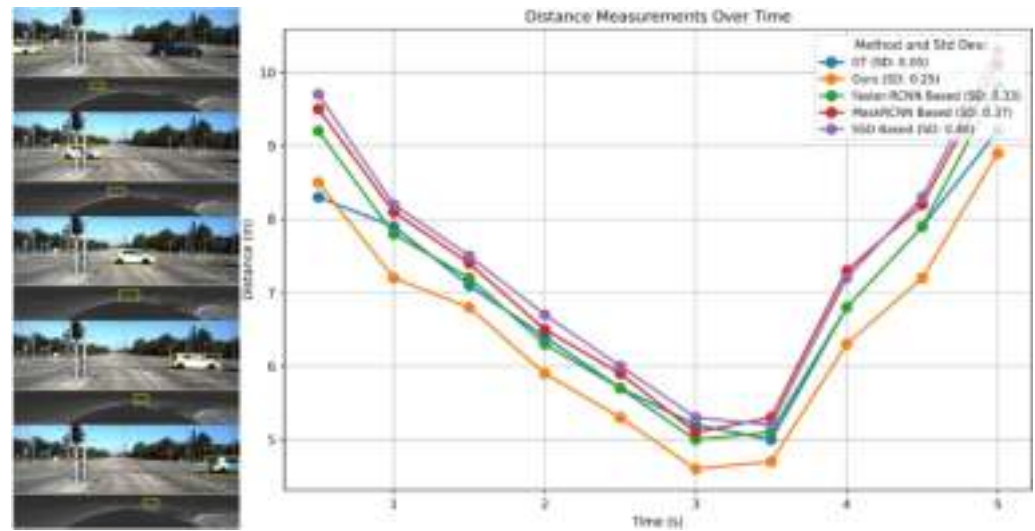


Figure 7. Statistical results and standard deviation calculations of different monocular distance measurement methods. The standard deviation of our method is 0.25. The standard deviation of Faster-RCNN is 0.33. The standard deviation of Mask R-CNN is 0.37. The standard deviation of SSD is 0.40.



Figure 8. The visualization results of different monocular distance measurement methods. (a) is the ground truth. (b) is the result based on Faster R-CNN. (c) is the result based on YOLOv5. (d) is the result based on YOLO-RedeCa.

4.5. Ablation Experiment

We adopted the ablation experiment method to evaluate the impact of different modules on the detection performance of YOLOv5m model on VOC dataset. The experimental results are shown in Table 6. We investigate the effect of depthwise separable convolution (DW) on model performance. In addition, we examine the impact of effective channel attention (ECA). We also explore the direct combination of DW and ECA. Finally, we assess the performance of the RedeCa module.

Table 6. Detection results using different modules on VOC dataset.

Method	DW	ECA	RedeCa (DW + ECA)	Precision (%)	mAP (%)	IOU (%)	Params (M)	FLOPS (G)
YOLOv5m				82.5	74.3	67.2	21.0	48.5
YOLOv5m	✓			70.5	62.9	38.4	9.1	18.3
YOLOv5m		✓		82.7	76.3	74.1	21.8	48.6
YOLOv5m	✓	✓		79.3	74.6	71.3	20.8	47.6
YOLOv5m + RedeCa			✓	81.9	77.5	73.4	17.2	34.6

As shown in Table 6, the baseline YOLOv5m model achieved a Precision of 82.5%, mAP of 74.3%, and IOU of 67.2% on the VOC dataset. It had 21.0 million parameters and 48.5 GFLOPS.

After adding Depthwise Separable Convolutions (DW) to the baseline model, Precision decreased to 70.5%, mAP dropped to 62.9%, and IOU significantly declined to 38.4%. However, the number of parameters and FLOPS were reduced to 9.1 million and 18.3 GFLOPS, respectively. This indicates that while the DW module effectively reduces model complexity, it negatively impacts detection performance.

Incorporating the Efficient Channel Attention (ECA) module resulted in a slight improvement in Precision to 82.7%, an increase in mAP to 76.3%, and a significant rise in IOU to 74.1%. The number of parameters and FLOPS slightly increased, demonstrating that the ECA module enhances detection performance without significantly increasing the computational burden.

Using Depthwise Separable Convolutions (DW) and Efficient Channel Attention (ECA) directly, the model saw slight improvements in Mean Average Precision (mAP) and Intersection over Union (IoU). But the increase was minimal, and there was no significant reduction in the network's parameters and computational load. However, by effectively integrating DW and ECA into the RedeCa module, the model's accuracy slightly decreased to 81.9%, but the mAP increased to 77.5%, and the IoU rose to 73.4%, while the parameters and FLOPS were reduced to 17.2 M and 34.6 G, respectively. This indicates that the use of the RedeCa module can balance detection performance and model complexity to some extent, providing an effective solution for real-time object detection applications.

To verify that the use of an anomaly jump filter helps smooth distance variations and eliminate potential errors caused by sudden changes in the detection bounding box size, we conducted a comparative analysis. We compared the algorithm architecture with the anomaly jump filter to the one without it.

As shown in Figure 9, using a Kalman filter integrated with the time series effectively eliminates potential errors caused by sudden changes in detection bounding box size. This reduces the dispersion in distance measurements, making them more consistent with the actual distances over time.

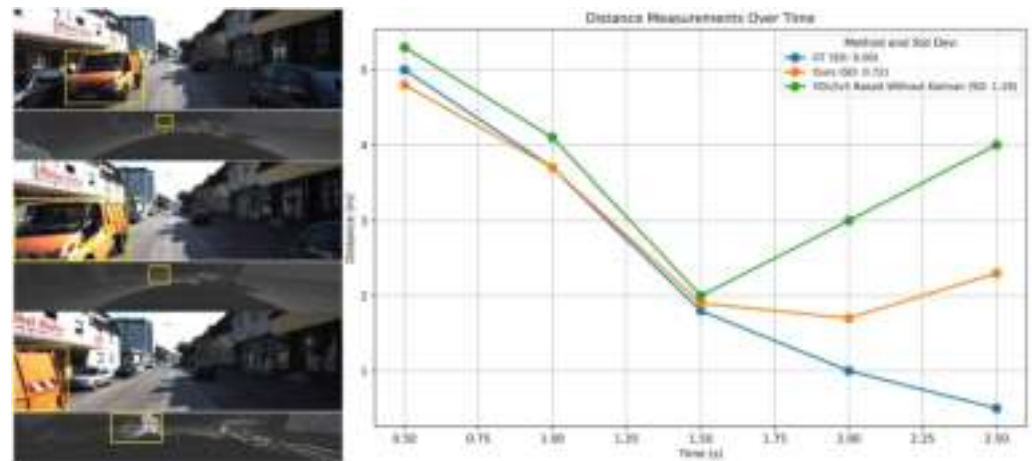


Figure 9. The impact of using Kalman filtering on the fluctuations in distance estimation. The standard deviation of our method is 0.72. The standard deviation for the YOLOv5-RedeCa-based method without the use of Kalman filtering is 1.29.

5. Conclusions

Our improved object detection network performs excellently on the COCO dataset. It achieves 71.6% precision, 61.5% mean Average precision (mAP@.5) and 57.9% recall, respectively. These results are better than the comparison models under the same computing resources. In addition, on the VOC dataset, our improved object detection network also shows excellent performance in mAP@.5 and recall. The results are 77.5% and 73.4%, respectively. By integrating the Bot-Sort algorithm, the robustness of the proposed method is significantly improved when dealing with object occlusion or temporary loss, so that the system can track and distance multiple objects at the same time. At the same time, the abnormal jump filter can effectively smooth the sudden change of object range, eliminate the potential anomaly caused by the mutation of the detection box size, and thus enhance the reliability of ranging.

The holistic algorithm architecture we designed and implemented successfully breaks through the traditional research gap between 2D object detection and monocular depth estimation. We combine the two techniques by designing a new algorithm architecture. This significantly improves the performance of object detection and distance estimation in dynamic and complex environments. The integration of these technologies not only optimizes the respective functions, but also improves the efficiency of the system in practical applications. In particular, it has shown great application potential in the field of traffic monitoring and automatic driving.

Author Contributions: Conceptualization, H.L., Y.M., Y.D. and Y.Y.; methodology, H.L. and Y.M.; software, H.L.; validation, H.L. and Y.Y.; formal analysis, H.L., Y.Y. and Y.M.; investigation, H.L.; resources, H.L., Y.D. and Y.M.; data curation, H.L.; writing—original draft preparation, H.L.; writing—review and editing, H.L., Y.M. and Y.Y.; visualization, H.L.; supervision, Y.D. and Y.M.; project administration, H.L.; funding acquisition, Y.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work is mainly supported by the Vehicle-road Cooperative Autonomous Driving Fusion Control Project, and sponsored by the Academic Research Projects of Beijing Union University (Nos. ZK80202003, ZK90202105), the Beijing Municipal Education Commission Science and Technology Program (Nos. KM202111417007, KM202211417006).

Data Availability Statement: The COCO dataset used can be obtained from <https://cocodataset.org/#home> (accessed on 24 June 2024). The VOC dataset can be obtained from <http://host.robots.ox.ac.uk/pascal/VOC/> (accessed on 24 June 2024). The KITTI dataset can be obtained from https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d (accessed on 24 June 2024). The fusion architecture using YOLO-RedeCa and abnormal jumping change filter is available

on GitHub at <https://github.com/Mr-Lv-BUU/The-Fusion-Architecture-Using-YOLO-RedeCa-and-Abnormal-Jumping-Change-Filter.git> (accessed on 24 June 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tan, K.; Wu, J.; Zhou, H.; Wang, Y.; Chen, J. Integrating Advanced Computer Vision and AI Algorithms for Autonomous Driving Systems. *J. Theory Pract. Eng. Sci.* **2024**, *4*, 41–48.
2. Haris, M.; Glowacz, A. Road object detection: A comparative study of deep learning-based algorithms. *Electronics* **2021**, *10*, 1932. [\[CrossRef\]](#)
3. Li, Y.; Huang, B.; Chen, Z.; Cui, Y.; Liang, F.; Shen, M.; Liu, F.; Xie, E.; Sheng, L.; Ouyang, W.; Shao, J. Fast-BEV: A Fast and Strong Bird's-Eye View Perception Baseline. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, 1–14.
4. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
5. Diwan, T.; Anirudh, G.; Tembhurne, J.V. Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimed. Tools Appl.* **2023**, *82*, 9243–9275. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Zhao, Z.Q.; Zheng, P.; Xu, S.T.; Wu, X. Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Poggi, M.; Tosi, F.; Batsos, K.; Mordohai, P.; Mattocchia, S. On the synergies between machine learning and binocular stereo for depth estimation from images: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 5314–5334. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Saxena, A.; Sun, M.; Ng, A.Y. Make3d: Learning 3d scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *31*, 824–840. [\[CrossRef\]](#)
9. Karsch, K.; Liu, C.; Kang, S.B. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2144–2158. [\[CrossRef\]](#)
10. Fu, C.; Yuan, H.; Xu, H.; Zhang, H.; Shen, L. TMSO-Net: Texture adaptive multi-scale observation for light field image depth estimation. *J. Vis. Commun. Image Represent.* **2023**, *90*, 103731. [\[CrossRef\]](#)
11. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *Proc. IEEE* **2023**, *111*, 257–276. [\[CrossRef\]](#)
12. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [\[CrossRef\]](#)
13. Xiao, Y.; Tian, Z.; Yu, J.; Zhang, Y.; Liu, S.; Du, S.; Lan, X. A review of object detection based on deep learning. *Multimed. Tools Appl.* **2020**, *79*, 23729–23791. [\[CrossRef\]](#)
14. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
16. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Martin, D.; Fowlkes, C.; Tal, D.; Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Proceedings of the Eighth IEEE International Conference on Computer Vision, ICCV 2001, Vancouver, BC, Canada, 7–14 July 2001.
18. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 379–387.
19. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv* **2013**, arXiv:1312.6229.
20. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
21. Han, X.; Chang, J.; Wang, K. Real-time object detection based on YOLO-v2 for tiny vehicle object. *Procedia Comput. Sci.* **2021**, *183*, 61–72. [\[CrossRef\]](#)
22. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
23. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
24. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
25. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Kwon, Y.; Michael, K.; Fang, J.; Wong, C.; Yifu, Z.; Montes, D.; et al. *ultralytics/yolov5: v6.2—YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.ai Integrations, v6.2*; Zenodo: Geneva, Switzerland, 2022.
26. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.

27. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.
28. Li, Z.; Yang, L.; Zhou, F. FSSD: Feature fusion single shot multibox detector. *arXiv* **2017**, arXiv:1712.00960.
29. Ren, D.; Yang, M.; Wu, J.; Zheng, N. Surface normal and Gaussian weight constraints for indoor depth structure completion. *Pattern Recognit.* **2023**, *138*, 109362. [[CrossRef](#)]
30. Favaro, P.; Soatto, S. A geometric approach to shape from defocus. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 406–417. [[CrossRef](#)] [[PubMed](#)]
31. Liu, B.; Gould, S.; Koller, D. Single image depth estimation from predicted semantic labels. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 1253–1260.
32. Eigen, D.; Puhersch, C.; Fergus, R. Depth map prediction from a single image using a multi-scale deep network. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2366–2374.
33. Kuznetsov, Y.; Stuckler, J.; Leibe, B. Semi-supervised deep learning for monocular depth map prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6647–6655.
34. Eigen, D.; Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2650–2658.
35. Liu, C.; Yang, G.; Zuo, W.; Zang, T. DPDFormer: A Coarse-to-Fine Model for Monocular Depth Estimation. *ACM Trans. Multimed. Comput. Commun. Appl.* **2024**, *20*, 1–21. [[CrossRef](#)]
36. Li, Z.; Chen, Z.; Liu, X.; Jiang, J. Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation. *Mach. Intell. Res.* **2023**, *20*, 837–854. [[CrossRef](#)]
37. Jiao, J.; Cao, Y.; Song, Y.; Lau, R. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 53–69.
38. Miclea, V.C.; Nedeveschi, S. Monocular depth estimation with improved long-range accuracy for UAV environment perception. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–15. [[CrossRef](#)]
39. Garg, R.; Bg, V.K.; Carneiro, G.; Reid, I. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part VIII 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 740–756.
40. Godard, C.; Mac Aodha, O.; Brostow, G.J. Unsupervised monocular depth estimation with left-right consistency. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 270–279.
41. Zhou, T.; Brown, M.; Snavely, N.; Lowe, D.G. Unsupervised learning of depth and ego-motion from video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1851–1858.
42. Mahaur, B.; Mishra, K. Small-object detection based on YOLOv5 in autonomous driving systems. *Pattern Recognit. Lett.* **2023**, *168*, 115–122. [[CrossRef](#)]
43. Aharon, N.; Orfaig, R.; Bobrovsky, B.Z. BoT-SORT: Robust associations multi-pedestrian tracking. *arXiv* **2022**, arXiv:2206.14651.
44. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
45. Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [[CrossRef](#)]
46. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [[CrossRef](#)]
47. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
48. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
49. Lou, H.; Duan, X.; Guo, J.; Liu, H.; Gu, J.; Bi, L.; Chen, H. DC-YOLOv8: Small-size object detection algorithm based on camera sensor. *Electronics* **2023**, *12*, 2323. [[CrossRef](#)]
50. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. YOLOv9: Learning what you want to learn using programmable gradient information. *arXiv* **2024**, arXiv:2402.13616.
51. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-time end-to-end object detection. *arXiv* **2024**, arXiv:2405.14458.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

An Encrypted Traffic Classification Approach Based on Path Signature Features and LSTM

Yihe Mei ¹, Nurbol Luktarhan ^{2,*}, Guodong Zhao ¹ and Xiaotong Yang ²¹ School of Software, Xinjiang University, Urumqi 830091, China; 107552204789@stu.xju.edu.cn (Y.M.); 107552204851@stu.xju.edu.cn (G.Z.)² College of Information Science and Engineering, Xinjiang University, Urumqi 830046, China; 107552204103@stu.xju.edu.cn

* Correspondence: nurbol@xju.edu.cn

Abstract: Classifying encrypted traffic is a crucial aspect of network security. However, popular methods face several limitations, such as a reliance on feature engineering and the need for complex model architectures to ensure effective classification. To address these challenges, we propose a method that combines path signature features with Long Short-Term Memory (LSTM) models to classify service types within encrypted traffic. Our approach constructs traffic paths using packet size and arrival times. We generate path signature features at various scales using an innovative multi-scale cumulative feature extraction technique. These features serve as inputs for LSTM networks to perform the classification. Notably, by using only 24 sequential packet features in conjunction with LSTM models, our method has achieved significant success in classifying service types within encrypted traffic. The experimental results highlight the superiority of our proposed method compared to leading approaches in the field.

Keywords: encrypted traffic classification; path signature features; long short-term memory; deep learning



Citation: Mei, Y.; Luktarhan, N.; Zhao, G.; Yang, X. An Encrypted Traffic Classification Approach Based on Path Signature Features and LSTM. *Electronics* **2024**, *13*, 3060. <https://doi.org/10.3390/electronics13153060>

Academic Editor: Martin Reisslein

Received: 27 June 2024

Revised: 29 July 2024

Accepted: 31 July 2024

Published: 2 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the era of ubiquitous Internet connectivity, network traffic classification has emerged as a cornerstone in cybersecurity. This discipline's importance is underscored by its critical applications in intrusion detection, traffic trend analysis, quality of service management, and enhancing network visibility [1]. With the widespread adoption of encryption technologies, the landscape of network traffic has dramatically changed as more users and service providers choose to transmit information through encrypted traffic [2].

Encrypted traffic differs from unencrypted traffic in that it employs encryption protocols to encode data transmitted between devices, thereby preventing unauthorized parties from accessing it. This encryption ensures significant advantages in terms of privacy and security by safeguarding sensitive information from interception and exploitation by malicious actors. Encrypted traffic maintains the confidentiality and integrity of user data, communications, and transactions, making them tamper-proof. In contrast, unencrypted traffic transmits data in plain text, which can be easily intercepted and read by anyone with network access. While unencrypted traffic allows for simpler monitoring and analysis by network administrators, it exposes users to a higher risk of data breaches and privacy violations. The primary advantages of unencrypted traffic are its simplicity and lower computational overhead, which can result in faster data transmission. However, the security benefits of encrypted traffic far outweigh these advantages, making encryption essential in today's digital environment.

Although encrypted traffic safeguards the privacy and anonymity of most users, it simultaneously presents substantial challenges to cybersecurity [3]. Criminals increasingly leverage sophisticated encryption techniques to conduct illegal activities, significantly

undermining the stability of cyberspace and posing threats to national security. Researchers urgently need new classification methods to address this formidable challenge.

The advent of encryption has posed a substantial challenge to traditional traffic classification methods [4]. Techniques that once relied on payload inspection and port-based classification have now been rendered ineffective against the veil of encryption [5,6].

Historically, the field has leaned on statistical methods and, more recently, machine learning algorithms to navigate the complexities of encrypted traffic classification [7–10]. These methods, however, come with their own set of challenges. They often require extensive feature engineering efforts and are heavily dependent on the quality and quantity of the available data. Moreover, although practical, the intricate designs of deep learning models demand vast computational resources and large datasets to achieve satisfactory accuracy [4].

Nowadays, real-world applications for encrypted traffic classification include Cisco Encrypted Traffic Analytics (ETA), Darktrace, Symantec Encrypted Traffic Management (ETM), and Vectra AI. Cisco ETA identifies malware in encrypted traffic using network telemetry and machine learning. Darktrace uses AI-driven anomaly detection to identify threats in encrypted data. Symantec ETM decrypts, inspects, and re-encrypts traffic, utilizing deep packet inspection for classification. Vectra AI's Cognito platform employs machine learning and behavioral analysis to detect threats within encrypted traffic. These solutions highlight the industry's reliance on machine learning, AI, and advanced network analysis for secure and effective encrypted traffic management.

The path signature feature proposed by Chen [11] has been successfully applied in recent years to fields such as machine learning, pattern recognition, and path analysis. This feature has also recently been introduced to the domain of encrypted traffic classification, where it has demonstrated its feasibility across several datasets [12]. However, the inherent complexity of path signature features means that classification results are not consistently satisfactory. In encrypted traffic classification, the challenge of fully leveraging path signature features to attain satisfactory results persists, necessitating further investigation and innovation.

In this paper, we introduce a multi-scale cumulative path signature feature extraction method to maximize the information content of path signature features. To fully exploit the potential of these features, we have explored their integration with deep neural networks. This integration has led to satisfactory classification results, demonstrating the effectiveness of combining path signature features with deep learning in encrypted traffic classification.

The main contributions of this paper are as follows:

1. We propose an encrypted traffic classifier that combines path signature features with an LSTM model. This classifier extracts the complex dynamics and geometric structures of encrypted traffic using path signature features and utilizes an LSTM model to perform the classification task, significantly reducing the computational resources required for traffic classification.
2. We introduce a multi-scale cumulative feature extraction method to generate path signature features that are optimally suited for LSTM models. Using this method, the classification accuracy of the LSTM model can be improved by 5–30%, validating the effectiveness of our new feature extraction approach.
3. Our proposed method achieves competitive classification results using only 24 consecutive packet length and time interval features. It demonstrates classification accuracies of 94.74%, 90.53%, 93.86%, and 95.03% on the ISCX-VPN, ISCX-nonVPN, ISCX-Tor, and ISCX-nonTor datasets, respectively, proving the effectiveness of our approach.

2. Related Work

2.1. Encrypted Traffic Classification

Internet traffic classification can be categorized into three main domains:

1. Distinguishing between encrypted and unencrypted traffic [3];
2. Identifying the specific applications associated with the traffic [13];

3. Classifying traffic based on different service types. Service types refer to the purpose of the traffic, i.e., traffic generated to meet specific user needs. For example, file transfer traffic is generated when users perform upload or download operations on the network, and streaming traffic is generated when users use streaming media services, such as listening to music or watching videos [14].

This section summarizes vital methods for network traffic classification, categorized based on the techniques used: port-based, payload-based, fingerprint construction, statistical methods, and deep learning models.

Port-based: This method, one of the oldest in network traffic classification, relied on fixed port numbers assigned by the Internet Assigned Numbers Authority (IANA) during the Internet's early development. This approach facilitated the association of network applications with their corresponding traffic via port scanning, thus enabling traffic identification [15]. However, with the exponential growth of network applications and dynamic port techniques, port scanning no longer applies to the current network environment [5]. Our method, notably, does not depend on port numbers.

Payload-based: Addressing the limitations of port scanning, researchers have proposed Deep Packet Inspection (DPI) methods [6] for traffic identification. DPI examines a packet's data part (payload) and uses known signatures to identify the traffic type. The core principle of DPI is that various communication protocol data packets have certain packet formats, and specific characters or strings with meaningful information can be found by extracting the payload of packets. These specific payload contents serve as indicators for traffic type recognition [6]. However, this method poses risks of infringing on user privacy, incurring high computation costs, and becoming ineffective when dealing with encrypted traffic [16]. Our approach circumvents the reliance on payload information.

Fingerprint construction: To address the limitations of payload-based traffic classification methods, several studies have utilized unencrypted protocol field information for traffic classification [3]. For instance, T Van Ede et al. [17] employs features such as device type, certificate details, packet size, and timing information to represent each flow, constructing a fingerprint library through clustering and cross-correlation techniques for efficient traffic classification. However, these methods heavily rely on plaintext information, which can be easily tampered with during transmission, leading to the loss of accurate meaning. Conversely, our model operates independently of plaintext information within the data packets.

Statistical methods: This category involves manually extracting features such as traffic size and timing using machine learning algorithms for traffic classification. The Random Forest classifier is a classic machine learning algorithm that operates by constructing multiple decision trees during training and outputting the class pattern for the classification task. This ensemble method enhances the model's robustness and accuracy by mitigating the common overfitting issue in individual decision trees. For example, VF Taylor et al. [9] uses a Random Forest classifier with packet size features for classification tasks. On the other hand, Vinai George Biju [18] employed the Support Vector Machine (SVM) algorithm to classify encrypted traffic. SVM is also a classic machine learning algorithm that works by finding the optimal hyperplane that maximizes the margin between different classes in the feature space, performing exceptionally well in high-dimensional spaces. Although these methods bypass the need for packet content inspection, thus suiting encrypted traffic classification, they also face limitations. Simple statistical features often fall short of accurate classification, and deriving complex features demands significant effort and resources. Our methodology relies on straightforward statistical features to derive path signature features, enriching feature informativeness without complex engineering.

Deep learning models: The advent of deep learning has significantly influenced the use of these models in encrypted traffic classification. DF [19] first uses a Convolutional Neural Network (CNN) to extract features from encrypted traffic, while FS-NET [20] employs a recursive neural network for encrypted traffic classification. DeepPacket [6] and TSCRNN [21] build models by combining feature extraction and classifier construction,

automatically handling feature extraction and classification when the payload of original packets is input into the model. Zhang et al. [22] used graph neural networks to extract packet-level features for traffic classification. MT-FlowFormer [23] uses a semi-supervised flow transformer to complete encrypted flow classification. Lin et al. [16] captured robust and general associations from raw traffic packets using large pre-trained models such as Bidirectional Encoder Representation from Transformers (BERT), achieving excellent classification results. Nonetheless, deep learning models typically require intricate network architectures and vast datasets for optimal performance. Contrarily, our model is designed to be efficient and requires fewer resources.

2.2. Path Signature Features

Path signatures were initially proposed by Chen [11] as a core concept in rough path theory, where “rough” refers to continuous paths that exhibit sharp fluctuations in rough paths. These paths are continuous but not differentiable everywhere, and the continuous packet size sequences in encrypted traffic can be seen as rough paths. In recent years, path signatures have regained researchers’ attention and have been successfully applied in machine learning, pattern recognition, and data analysis fields. Graham [24] used sliding-window-based path signature features with a CNN for handwritten character recognition, achieving excellent results. L. G. Gyurkó [25] used path signatures to extract features from financial data flows for financial data classification. I. P. Arribas [26] applied path signatures to the medical field to differentiate mental disorders. In [12], a novel methodology was introduced for describing signatures through length-normalized path signatures. The study also explored path signature features, employing them in conjunction with the Random Forest algorithm for the classification of encrypted traffic. Overall, these applications demonstrate the effectiveness of extracting features using path signature functions and the value of representing information-rich vectors. The next section introduces the definition of path signatures, their low-order geometric interpretation, and relevant properties. For more mathematical proofs, refer to [11].

2.2.1. Definition of Path Signatures

A path signature is a transformative mapping function that converts raw path data into a sequence of real numbers [12]. Each number within this sequence is derived uniquely through the data points along the original path, capturing a distinct geometric characteristic of that path. For a path X in d -dimensional Euclidean space $\mathbb{R}^d$, defined as a continuous mapping from the interval $[a, b]$ to $\mathbb{R}^d$, written as $X: [a, b] \rightarrow \mathbb{R}^d$, the parameter $t \in [a, b]$ delineates a trajectory within this d -dimensional Euclidean space:

$$X_t = \{X_t^1, X_t^2, X_t^3, \dots, X_t^d\}. \quad (1)$$

where X_t^d represents an arbitrary continuous path in the d -th dimension. The path signature is an infinite sequence of a path $X: [a, b] \rightarrow \mathbb{R}^d$, consisting of the zeroth-order signature (1) and various order signature terms (also called iterated integrals):

$$S(X)_{a,b} = (1, S(X)_{a,b}^1, \dots, S(X)_{a,b}^d, S(X)_{a,b}^{1,1}, S(X)_{a,b}^{1,2}, \dots, S(X)_{a,b}^{d,d} \dots) \quad (2)$$

For a d -dimensional path X , its k -fold iterated integral can be expressed as follows:

$$S(X)_{a,t}^{i_1, \dots, i_k} = \int_{a < t_k < t} \dots \int_{a < t_1 < t_2} dX_{t_1}^{i_1} \dots dX_{t_k}^{i_k}. \quad (3)$$

In Formula (3), $t_1, t_2, \dots, t_k$ refer to integration variables that are distinct from t . This distinction is made to avoid confusion with the path parameter t during multiple integrations. Taking the two-dimensional path $X_t = \{X_t^1, X_t^2\}$ shown in Figure 1 as an example, its third-order path signature features are as follows: {57.17, 54.67, 23.67, 30.33, 44.17, 34.33, 22.83, 20.83}.

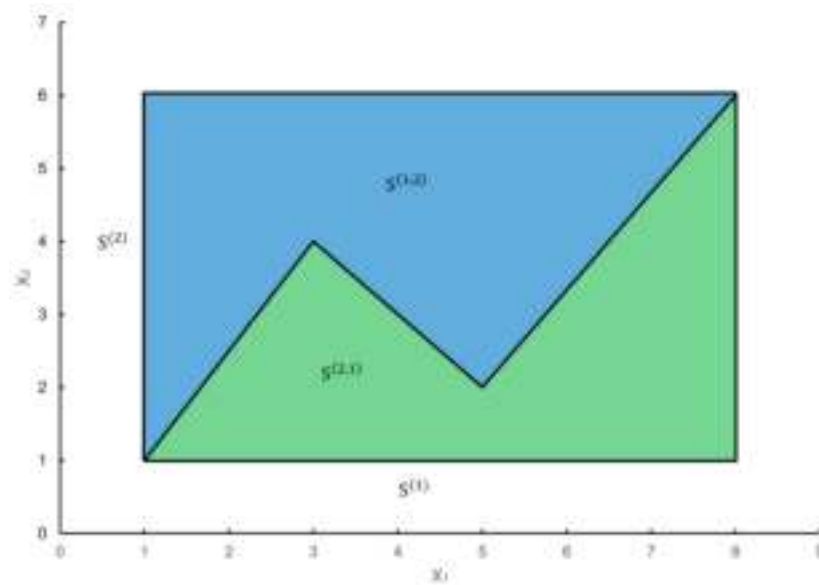


Figure 1. Geometric interpretation of two-dimensional path.

2.2.2. Geometric Interpretation of Path Signatures

To further explain the geometric interpretation of path signature features, we use a two-dimensional path $X_t = \{X_t^1, X_t^2\}$ as an example, where $t \in [0, 4]$, ($a = 0, b = 4$). The path is defined by $X_t^1 = \{1, 3, 5, 8\}$ and $X_t^2 = \{1, 4, 2, 6\}$. It is important to note that t is a continuous parameter; for the sake of convenience, we have chosen four discrete points to construct the path. The shape of the path is shown in Figure 1. The computed zeroth-order, first-order, and second-order path signatures are as follows:

$$S = \{1, S^{(1)}, S^{(2)}, S^{(1,1)}, S^{(1,2)}, S^{(2,1)}, S^{(2,2)}\} = \{1, 7, 5, 24.5, 19, 16, 12.5\}. \quad (4)$$

Within the path signature framework, the first element, denoted by 1, is a constant, signifying the zeroth-order characteristic of the path. $S^{(1)}$ and $S^{(2)}$ represent the first-order path signature components, corresponding to the projection distances of the path along the X_1 and X_2 axes, respectively. The expressions $S^{(1,2)}$ and $S^{(2,1)}$ denote the areas bounded by the path and the straight lines parallel to the coordinate axes, extending through the path's start and end points. Meanwhile, $S^{(1,1)}$ and $S^{(2,2)}$ reflect half the squared projection distances of the path on the X_1 and X_2 axes, correspondingly. We explain only the zeroth-, first-, and second-order path signature features, as higher-order path signatures are difficult to describe visually, and our method uses only up to second-order signatures.

2.2.3. Properties of Path Signatures

In this section, we will elaborate on the crucial properties of path signature features. These properties constitute the theoretical foundation and notable advantages of their application in encrypted traffic classification. They are detailed as follows:

- **Uniqueness:** Hambly et al. [27] established that each rough path possesses a distinct path signature, ensuring a one-to-one correspondence between non-tree-like paths and their signatures. This fundamental property asserts that path signatures can precisely encapsulate the geometric traits of paths. Incorporating time as a monotonically increasing dimension in the sequence of encrypted traffic transforms these sequences into non-tree-like paths, thus providing a robust theoretical framework for substituting original paths with their path signature features as input for analytical models.
- **Invariance under parameter changes:** Different sampling strategies yield varied parameters for the same path, yet the path signature remains consistent across these variations [27]. This invariance implies that classification outcomes for traffic from

specific application types remain stable, unaffected by the diversity in parameters. Leveraging this attribute allows for the elimination of discrepancies introduced by various reparameterizations of traffic within the same category, highlighting a crucial advantage of utilizing path signature features.

- Dimension invariance: The dimensionality of path signature features is determined solely by the chosen truncation depth, independent of the actual path length [27]. For instance, considering the previously discussed two-dimensional path $X_t = \{X_t^1, X_t^2\}$, $t \in [0,4]$, ($a = 0, b = 4$) with $X_t^1 = \{1, 3, 5, 8\}$ and $X_t^2 = \{1, 4, 2, 6\}$, truncating to one dimension results in a path signature length of 3, while truncation to two dimensions yields a length of 7. This fixed-length feature extraction from paths of varying lengths significantly simplifies the feature extraction process, especially for models necessitating fixed-length input features.

2.3. LSTM Model

LSTM networks, a subclass of Recurrent Neural Networks (RNNs), are distinguished by their capacity for maintaining long-term dependencies [28]. Shen et al. used LSTM to analyze the drag coefficient of car bodies [29]. Ref. [30] applied LSTM to predict groundwater levels, showcasing the model's ability to handle fluctuating environmental data and provide accurate forecasts. Ref. [31] explored the prospects of applying LSTM in the field of autonomous driving, highlighting its potential to process and learn from vast amounts of sequential data. These diverse applications underscore the versatility of LSTM in various fields and its robust capability in handling time series data. Illustrated in Figure 2, LSTM units are set apart from traditional RNNs through the integration of three distinct “gates”, which collectively enhance the model’s memory management capabilities [32]:

- Input Gate: Oversees the flow of incoming data into the cell, deciding how much of the new information should be stored;
- Output Gate: Determines the extent to which the cell’s current state influences other parts of the network, controlling the output flow;
- Forget Gate: Adjusts the cell’s self-recurrent connections using sigmoid functions to scale values between 0 and 1, determining what information is discarded or retained.

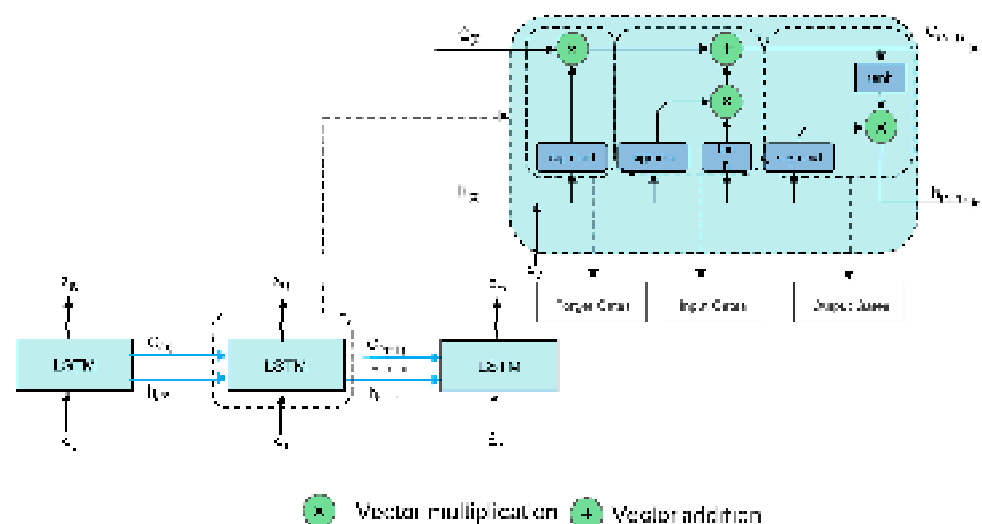


Figure 2. LSTM model architecture.

As a typical time series model, LSTM effectively handles path signature features within sequences. Known for its ability to capture long-term dependencies in sequential data [29–31], LSTM is particularly well-suited for this task due to the inherent sequential nature of network traffic data. Compared to other well-known deep learning models such as CNNs or Transformers, LSTM has the following advantages in this encrypted traffic classification task:

- Better handling of temporal dependencies: While CNNs excel at capturing spatial hierarchies and patterns, they are typically more effective in processing data with spatial correlations (such as images) and cannot capture temporal dependencies as effectively as LSTM [33].
- More straightforward and more direct data processing: Applying CNNs to sequence data requires transforming the time series into a format suitable for convolution operations, necessitating more complex data processing. In contrast, LSTM allows direct input of time series features into the model.
- Lower computational resource requirements: Transformer models have recently gained popularity for their success in sequence-to-sequence tasks, especially in natural language processing. Transformers use self-attention mechanisms to capture dependencies between different parts of the input sequence, regardless of their distance [34]. While this is beneficial for capturing global dependencies, Transformers typically require substantial computational resources and large datasets for practical training. LSTM networks, using inherent gating mechanisms to process sequential data, require fewer computational resources for training and deployment.

3. Methodology

3.1. Overview of the Approach

The process flow of our method for classifying encrypted traffic is illustrated in Figure 3. Initially, during the data preprocessing phase, we dissect pcap files from publicly accessible datasets by employing the five-tuple criteria, subsequently discarding packets lacking valid payloads. This step ensures the creation of data flows of uniform length. Moving forward to the feature extraction phase, we derive size, direction, and arrival time attributes from these uniform data flows to establish traffic paths. A transformation process is applied to these traffic paths to enrich them with additional information. The next stage involves leveraging multi-scale accumulation techniques to distill path signature features from these enhanced traffic paths. After extracting these features, we apply data-balancing techniques to achieve a well-balanced training set. These optimized path signature features are input into an LSTM model and fully connected layers to execute the final classification task. The forthcoming subsections will delve into each phase of this methodology in greater depth.

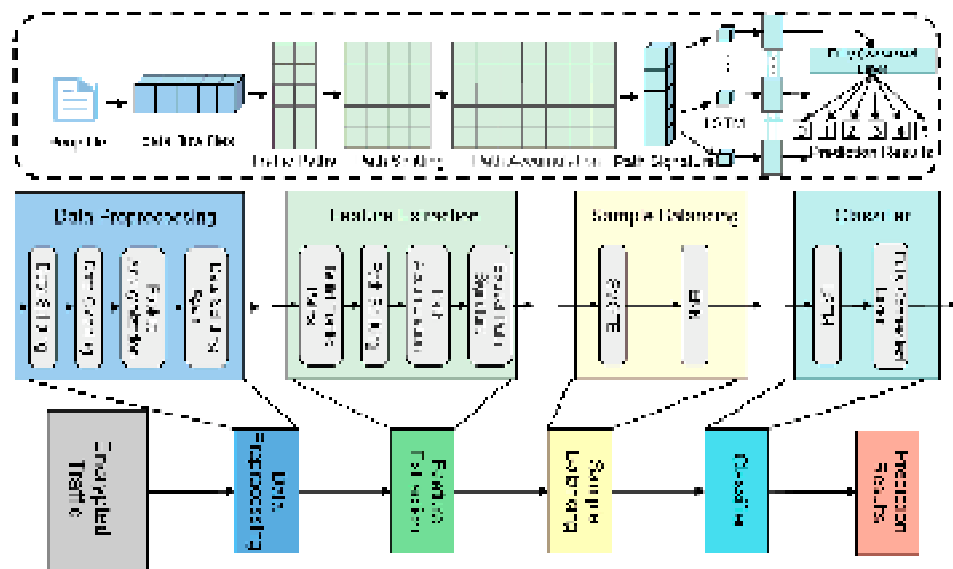


Figure 3. The overall framework of our approach.

3.2. Data Preprocessing

In handling the raw pcap files from the dataset, we employ SplitCap to segregate them into individual session files, utilizing the five-tuple criteria—source IP, source port, destination IP, destination port, and the transport layer protocol [3]. The following steps involve eliminating sessions devoid of data and duplicates. To mitigate the risk of model overfitting with this dataset, we anonymize the MAC and IP addresses in the packet data by substituting them with zeros [6]. Following this, we further refine the session files to ensure each contains precisely 24 data packets—a parameter whose significance will be elaborated on in the upcoming experimental section. Sessions not meeting the 24 data packet criterion are excluded, generating sample files that align with our experimental prerequisites.

3.3. Build Traffic Paths

Upon standardizing the session files to a uniform length, we categorize data packets transmitted from the client to the server with positive values while labeling packets moving in the reverse direction with negative values. This process results in the creation of a data packet length sequence, $L = (l_1, l_2, \dots, l_n)$, where l represents the length of each data packet [26]. Additionally, we capture the arrival time for each data packet. To synchronize these timestamps, we normalize the data to compile a time sequence for the packets, denoted as $T = (t_1, t_2, \dots, t_n)$, where t represents the arrival time for each data packet.

3.4. Transform Traffic Paths

Following the extraction of two-dimensional path signature features, which encapsulate the data packet time and length, we recognize that the representation of inter-session information remains incomplete. To enhance the utilization of inter-session characteristics, we adopt a methodology inspired by the work of Xu et al. [12]. This approach involves transforming the original traffic paths to unveil additional layers of information, thereby facilitating a more detailed and informative analysis.

3.4.1. Path Splitting

In our method, the term “client” refers to users who require Internet services. It is the clients’ use of various services on the Internet that generates different types of encrypted traffic. For instance, when a user uploads or downloads files online, file transfer traffic is produced. Similarly, when a user uses streaming services, such as listening to music or watching videos, streaming traffic is generated. To more accurately depict the traffic flow dynamics between the client and server, we partition the data packet length sequence into two distinct sequences. One sequence captures data transmissions from the client to the server, while the other details transmissions from the server to the client [12]. This division is formalized as follows below.

For client-to-server transmissions:

$$C = (c_1, c_2, \dots, c_n), c_i = \begin{cases} 0, & l_i < 0 \\ l_i, & l_i > 0. \end{cases} \quad (5)$$

Conversely, for server-to-client transmissions:

$$S = (s_1, s_2, \dots, s_n), s_i = \begin{cases} l_i, & l_i < 0 \\ 0, & l_i > 0. \end{cases} \quad (6)$$

By assigning data packet sizes from the opposite direction of transmission as zero, we underscore the distinctive flow properties inherent to each direction, laying a foundational framework for further path transformations.

3.4.2. Path Accumulation

Accumulation features play a crucial role in classifying encrypted traffic [35], primarily because the payload size tends to remain constant across the same type of traffic

service, leading to similar accumulation patterns. To leverage this consistency for revealing information within encrypted traffic payloads, we define the accumulation features as follows below.

Direct accumulation of packet lengths:

$$L' = (l'_1, l'_2, \dots, l'_n), l'_i = \sum_{i=1}^k l_i, \quad (7)$$

For client-directed transmissions:

$$C' = (c'_1, c'_2, \dots, c'_n), c'_i = \sum_{i=1}^k c_i, \quad (8)$$

For server-directed transmissions:

$$S' = (s'_1, s'_2, \dots, s'_n), s'_i = \sum_{i=1}^k s_i. \quad (9)$$

Hence, we achieve a comprehensive representation of the traffic path through the following:

$$X_t^{l,c,s,l',c',s',t} = \{L, C, S, L', C', S', T\}, t \in [0, n]. \quad (10)$$

This representation integrates the initial data packet length sequences and their accumulated versions alongside the arrival time sequence, offering a multifaceted view of the traffic flow for adequate classification. Taking a specific traffic path $L = [194, -83, -32, 53, 86]$ and $T = [0, 2, 5, 8, 10]$ as an example, the corresponding transformed traffic path is shown in Table 1.

Table 1. Examples of seven different traffic paths.

$X_5^{l,c,s,l',c',s',t}$						
L	C	S	L'	C'	S'	T
194	194	0	194	194	0	0
−83	0	−83	111	194	−83	2
−32	0	−32	79	194	−115	5
53	53	0	132	247	−115	8
86	86	0	218	333	−115	10

3.5. Extracting Path Signature

Upon the transformation of traffic paths, our next step involves extracting path signature features. We define the path feature f for each transformed path as follows:

$$f_t = \phi(X_t^{l,c,s,l',c',s',t}), t \in [0, n]. \quad (11)$$

Here, $\phi(X)$ denotes the function responsible for extracting the path signature and f_t represents the path signature of the first t data packets. It is important to note that the dimensionality of the path signature features depends solely on the path's chosen truncation depth, independent of the actual length of each path. Within a many-to-one LSTM model, it is customary to utilize the features of the last time step as the input for final classification.

Leveraging the unique properties of the path signature and the LSTM model, as depicted in Figure 4, we introduce a novel multi-scale accumulation feature extraction approach. This method enhances the informational value extracted at the final time step input to the model. The formulation and structure of the input feature sequence are detailed as follows:

$$F = (f_2, f_3, \dots, f_n). \quad (12)$$

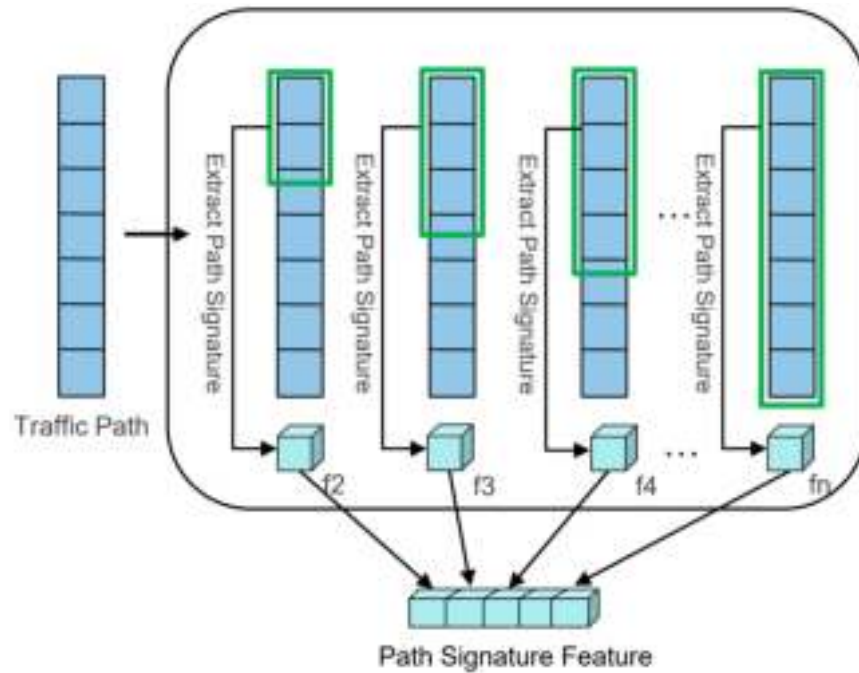


Figure 4. Multi-scale accumulation feature extraction method.

The requirement that $n \geq 2$ stems from the consideration that a path signature based on fewer than two data packets lacks substantive relevance. When $n = 1$, the path signature reduces to merely reflecting a single data packet's length and arrival time, failing to satisfy the criteria for our experimental setup. Consequently, we compile the path signature sequence F , where each item contains the path signature features of the first n data packets, and all path signatures have the same dimension.

3.6. Data Balancing

Encountering imbalanced data across different classifications is a common issue when collecting experimental data from datasets. Implementing data-balancing techniques is pivotal to bolster training tasks' effectiveness. This section delineates the strategies we have adopted to achieve data balance.

3.6.1. SMOTE (Synthetic Minority Over-Sampling Technique)

Introduced by Chawla [36], SMOTE is a leading oversampling methodology to rectify data imbalance issues. It augments the dataset by creating new, synthetic instances for the minority class derived from existing observations. Within the feature space, SMOTE crafts synthetic samples resembling the minority class. Specifically, for each minority class sample, SMOTE identifies its n nearest neighbors using Euclidean distance. Following this, one neighbor is chosen randomly, generating a vector linking this neighbor and the sample. A synthetic instance is then formed by adding a random fraction of this vector to the original sample. This procedure is repeated until the target oversampling rate is attained [36]. The SMOTE algorithm's formula is given as follows:

$$n = s + d * (s^R - s), \quad 0 \leq d \leq 1. \quad (13)$$

where s symbolizes the minority class samples, s^R denotes the neighbors determined by k nearest neighbors, d is a randomly selected value within the range $[0,1]$, and n represents the generated synthetic sample. As shown in Figure 5, two different colored points represent two different types of samples, the three nearest neighbors of sample s are s_1^R , s_2^R , and s_3^R . Using the SMOTE method, a neighbor s_1^R is randomly selected for sample s , and a new sample n is obtained through calculation.

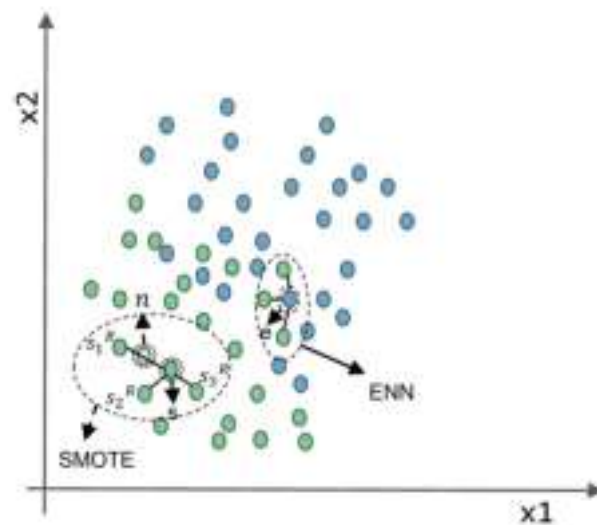


Figure 5. The example of SMOTE and ENN.

3.6.2. ENN (Edited Nearest Neighbors)

ENN employs the k nearest neighbors algorithm as a secondary sampling strategy [37]. It aims to prune the dataset by removing instances likely to be misclassified. ENN computes the Euclidean distance for every instance to pinpoint its k nearest neighbors (where k is commonly set to 3). If most of these neighbors do not share the instance's class, the instance and its neighbors are excised from the dataset [37]. Thus, ENN can effectively decrease the presence of majority class samples within the training set. Taking sample e in Figure 5 as an example, its k (set to 3) neighbors are all from other classes. Using the ENN method, all four samples will be removed.

3.6.3. SMOTE-ENN

SMOTE-ENN combines the strengths of both SMOTE and ENN into a unified approach to data balancing, as proposed by Batista et al. [38]. Initially, the dataset undergoes an enhancement via the SMOTE algorithm, which amplifies the minority class's representation through synthetic sample creation. Once an adequate number of minority class samples is produced, the ENN algorithm is applied to refine the dataset further by removing instances close to the classification boundary. This sequential application of SMOTE and ENN facilitates a more evenly distributed sample set across classes [38].

3.7. Input to the LSTM Model

The sequence of path signatures, denoted as F , serves as the input to the LSTM model, setting the stage for the classification process. The configuration of time steps within the LSTM correlates directly with the length of the initial traffic path, reduced by one. Consequently, the feature f_{n+1} at the n th time step encapsulates the path signature of a traffic path comprising $n + 1$ data packets. This arrangement ensures that with each additional time step, incrementally more information is incorporated, enriching the data representation. Notably, the feature at the final time step integrates the path signature of the entire traffic path, embodying the most exhaustive information available.

The inherent memory capability of the LSTM model [28] plays a pivotal role, enabling each time step to harness and convey information from preceding steps. As a result, the concluding time step effectively aggregates information across the entire sequence, circumventing the potential drawbacks of relying solely on global features. This comprehensive information integration significantly boosts the model's classification precision.

3.8. Fully Connected Layer

Following the LSTM's processing of the input sequence, the last time step's output undergoes extraction and is then directed towards a fully connected layer, where the classification endeavor is actualized. The essence of this layer lies in its capacity to transition the LSTM-acquired data features into the domain of label space. The operation within the fully connected layer is mathematically depicted as follows:

$$Y = W \times X + B. \quad (14)$$

where Y is the output vector, W is the weight of the fully connected layer, X is the input vector derived from the LSTM's final time step, and B is the bias.

3.9. Complete Workflow

In this section, we will explain our proposed method's workflow in detail. After capturing a traffic segment, we first extract the two basic statistical features of packet length and packet arrival time to form the traffic path. We then split the packet length sequence into two paths representing the direction of packet transmission. Next, we perform an accumulation process to obtain three paths representing the cumulative packet length transmission. This results in a seven-dimensional traffic path.

By employing the path signature feature function, we transform the seven-dimensional traffic path into high-dimensional path signature features, which are then fed into the LSTM model. The LSTM model processes the input sequence progressively, step by step. At each time step, the input features interact with the hidden state and memory cell state from the previous time step. Through a series of gating mechanisms, the hidden state and memory cell state are updated at each time step, allowing the model to effectively capture temporal dependencies within the traffic data.

After processing the entire sequence, the hidden state of the LSTM model at the final time step is selected as the feature output. This hidden state has integrated information from the entire input sequence and represents the global characteristics of the sequence data. The selected hidden state is then input into a fully connected layer. The fully connected layer, through a series of weights and activation functions, further processes the hidden state to generate a probability distribution for the classification result. Table 2 illustrates the changes in shape at each stage of processing an encrypted traffic sequence containing 24 packets.

Table 2. Data shape changes at each processing stage.

Stage	Description	Shape
Raw Data	Capture encrypted traffic sequence	—
Build Traffic Paths	Extract packet length and arrival time	(24, 2)
Path Splitting	Split the length sequence into two paths	(24, 4)
Path Accumulation	Accumulate three-length sequences	(24, 7)
Extracting Path Signature	Use path signature feature function to extract path signature features	(24, 56)
LSTM Input	Use path signature features as input to LSTM	(1, 24, 56)
LSTM Hidden Units	64 hidden units of LSTM layer process input sequence, returning hidden states for all time steps	(1, 24, 64)
Final Hidden State	Select the hidden state of the last time step in the LSTM model and input the fully connected layer	(1, 64)
Fully Connected Layer Output	The fully connected layer outputs the probability distribution of the classification result	(1, 6)

4. Discussion

In this section, we will elaborate on various aspects of our experimental framework, shedding light on various facets such as the experimental environment, evaluation metrics, datasets employed, data processing techniques, and the selection of hyperparameters. Furthermore, to establish our methodology's efficacy and competitive edge, we conducted comparative experiments against prevailing mainstream techniques. Ablation studies were also undertaken to underscore the contribution and impact of individual components within our proposed framework.

4.1. Experimental Environment

The experiments were conducted on a Windows 10 operating system with Python version 3.9. We implemented the LSTM model using PyTorch version 1.13.1. We set the initial learning rate for model training to 0.0001 and the training epochs to 500. The LSTM model used for this experiment has 64 hidden dimensions, and the output dimension of the subsequent fully connected layer is 6, corresponding to the six-class classification task of encrypted traffic service types.

4.2. Evaluation Metrics

We employed four evaluation metrics to assess the performance of the method: accuracy (AC), precision (PR), recall (RC), and F1 score (F1). True positives (TPs) are correctly identified positive samples, false positives (FPs) are negative samples misclassified as positive, true negatives (TNs) are correctly identified negative samples, and false negatives (FNs) are positive samples misclassified as negative.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (15)$$

$$Precision = \frac{TP}{TP + FP}, \quad (16)$$

$$Recall = \frac{TP}{TP + FN}, \quad (17)$$

$$F1\text{-score} = \frac{2 * Precision * Recall}{Precision + Recall}. \quad (18)$$

These metrics offer a comprehensive view of the model's efficacy, encapsulating its accuracy, precision, and sensitivity in classifying the data into its respective categories.

4.3. Datasets

To comprehensively verify the performance of our proposed method, we used two publicly available datasets: the ISCX VPN-nonVPN dataset [39] and the ISCX Tor-nonTor dataset [40]. These datasets were published by the University of New Brunswick.

The ISCX VPN-nonVPN dataset consists of two parts: ISCX-VPN and ISCX-nonVPN. The ISCX-VPN dataset contains traffic captured in a VPN environment for six service types: chat, email, file transfer, streaming, P2P, and VoIP. The ISCX-nonVPN dataset includes the same types of traffic, but captured outside the VPN environment.

Similarly, the ISCX Tor-nonTor dataset is divided into ISCX-Tor and ISCX-nonTor. The ISCX-Tor dataset collects traffic through the Tor network, which provides a high level of anonymity. The ISCX-nonTor dataset contains traffic collected outside the Tor environment, with traffic types identical to those in the ISCX-VPN dataset. Table 3 presents the main applications corresponding to each service type and provides a brief description.

Table 3. Dataset details.

Service	Application	Description
Chat	AIM, ICQ, Skype, Facebook, Hangouts	Traffic generated during online chat communication
Email	Email, Gmail	Traffic generated during email transmission
File transfer	Skype SFTP, FTPS, SCP	Traffic generated during file uploads and downloads
Streaming	Vimeo, YouTube, Netflix, Spotify	Traffic generated when using streaming applications
P2P	uTorrent, BitTorrent	Traffic generated when sharing torrent resources using P2P programs
VoIP	Facebook, Skype, Hangouts, VoipBuster	Traffic generated during online voice calls

After obtaining the dataset, we used the packet length and arrival time to construct our path signature features. Packet length refers to the byte size of each packet in the data stream. Since traffic transmission in a network is usually bidirectional, we added positive and negative signs to indicate the direction of the traffic: traffic from the client to the server was set as positive, and traffic from the server to the client was set as negative. This created a packet length sequence that represents both the transmission direction and packet load size.

Packet arrival time refers to the time at which each packet is received, starting from the reception of the first packet and recording the arrival time of each subsequent packet. This results in an increasing time series that represents the transmission frequency of the packets. By segmenting and accumulating the packet length sequence and normalizing the packet arrival times, we obtained the final traffic path. This path was then input into the path signature feature function to derive the traffic path signature features required for our method.

4.4. Data Balancing

In addressing the sample imbalance inherent in the ISCX VPN-nonVPN dataset for the classification of encrypted traffic, our methodology employed a multifaceted approach to ensure equitable representation across different traffic service types. Initially, a threshold of 5000 samples was established, whereby categories exceeding this number had a random subset of 5000 samples chosen, whereas categories falling below this threshold utilized all available samples to maximize data usage. After this selection, the category with the lowest sample count was identified, establishing a benchmark for uniformity. A portion corresponding to one-fifth of this minimal count was extracted from each category to assemble a comprehensive test set, ensuring a balanced representation. The SMOTE-ENN was applied to address the residual imbalance in the training dataset, synthesizing new samples for minority classes and removing ambiguous samples to refine the dataset quality. Figure 6 compares the number of samples for each category before and after balancing for the training set data in the ISCX-VPN dataset.

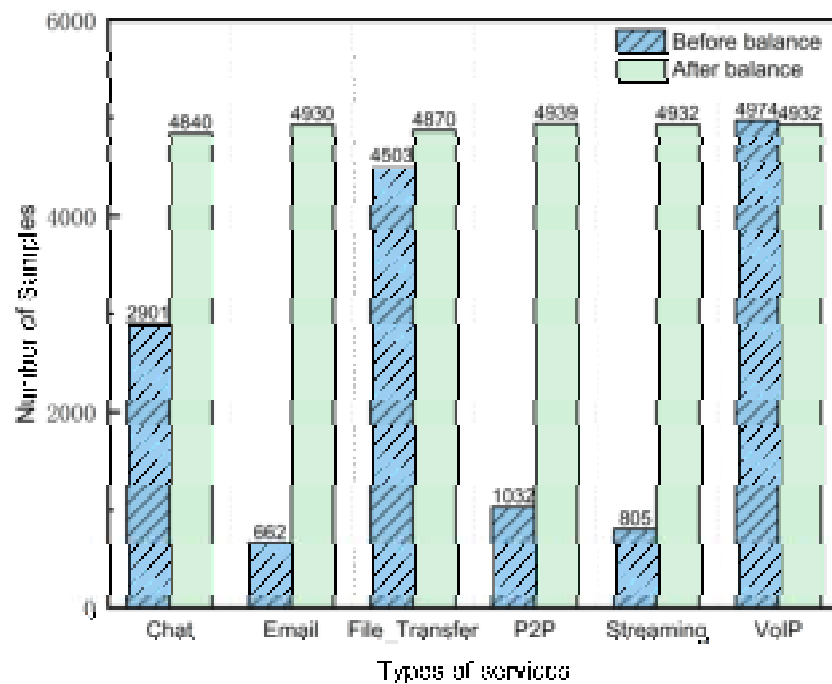


Figure 6. The number of samples.

4.5. Parameter Selection

In our experiments, the values of two crucial hyperparameters, namely the length of traffic paths and the dimensions of the path signatures, played a significant role and were closely related to the model's performance. They directly shaped the model's ability to learn from and interpret the data, influencing the analysis's depth and breadth. The traffic path length determines the input data for each analysis instance, impacting the model's comprehension of session continuity. Meanwhile, the dimensionality of the path signatures affects the complexity and detail of the features extracted from the traffic paths, balancing between information richness and computational feasibility. In this section, we elaborate on the process of obtaining these hyperparameters.

The optimal traffic path length is dictated by the necessity to include a sufficient number of consecutive data packets with valid payloads in a session file, which constructs the basis of the original traffic path. A path length that is too succinct may lead to session files devoid of adequate data for efficacious traffic classification tasks. Conversely, overly elongated traffic paths may unnecessarily increase data collection requirements without proportionate gains in classification accuracy [41]. Discerning the optimal length of traffic paths was vital as a foundational element in our experimental design.

In classifying service types for encrypted traffic under VPN conditions, we aimed to determine the optimal path length. As shown in Figure 7, when the path length was relatively short, the classification accuracy fluctuated and gradually improved with increasing path length for any given dimension. When the path length reached 20, various-dimensional models achieved relatively good classification performance. Among these, the two-dimensional path model attained the highest classification accuracy (94.74%) when the path length was 24. Subsequently, as the path length increased, the classification accuracy did not significantly improve and, in some dimensions, even decreased. Consequently, we selected 24 as our optimal parameter for the path length.

Moreover, the choice of path signature dimensions is equally critical. Higher-order path signatures inherently encapsulate more information than their lower-order counterparts. Nevertheless, a rise in path signature dimensions precipitates an exponential surge in feature dimensions [11]. For instance, a second-order signature feature in a seven-dimensional traffic path would have a dimensionality of 56, escalating to 399 for the third order and 2800 for the fourth order. This trend underscores the balance required

between the informational gain from increased path signature dimensions and the resultant computational complexity.

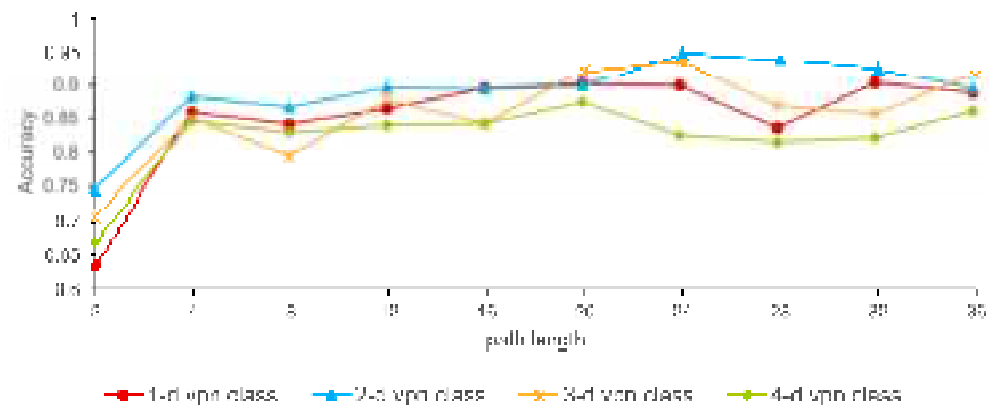


Figure 7. Effect of packet length and dimension (accuracy).

As shown in Figure 7, based on our experiments, increasing the dimensions did not significantly improve the classification performance. There was a slight decrease in performance when the dimensions were increased to four. We opted for a two-dimensional path signature as the optimal parameter considering classification performance and computational efficiency.

4.6. Sensitivity Analysis

To evaluate the robustness of our method, we conducted a sensitivity analysis. Sensitivity analysis is a technique used to assess a model's responsiveness to parameter changes. We observed the impact on model performance by varying the learning rate and regularization parameters during model training. Expressly, while evaluating the impact of the learning rate, we set the L2 regularization parameter to 0.0001, and while assessing the impact of the regularization parameter, we set the learning rate to 0.0001. The experimental results are shown in Tables 4 and 5.

Table 4. Classification accuracy under different learning rates.

Learning Rate	0.1	0.01	0.001	0.0001	0.00001
Accuracy	0.8596	0.8918	0.9152	0.9474	0.9298

Table 5. Classification accuracy under different regularization parameters.

Regularization Parameters	0.01	0.001	0.0001	0.00005	0.00001
Accuracy	0.8099	0.9006	0.9474	0.921	0.9152

It can be observed that as the learning rate and regularization parameters changed, our proposed method's classification accuracy remained consistently high, despite some fluctuations. This indicates the strong robustness of our method. The ability to maintain a high level of accuracy under varying conditions demonstrates the reliability and stability of our approach.

4.7. Ablation Study

To underscore the contributions of each component within our proposed method for encrypted traffic classification, we executed a series of ablation studies. These studies were designed to assess the impact of the path signature feature individually, the multi-scale cumulative feature extraction method, and the LSTM model on the overall performance.

Path signature feature ablation: For this part of the study, we no longer extract path signature features from the traffic paths. Instead, the raw seven-dimensional feature samples were fed directly into the LSTM model, bypassing the path signature processing step. All other experimental conditions were maintained using the established method.

Feature extraction method ablation: In evaluating the role of our multi-scale cumulative feature extraction method, we set aside this advanced approach in favor of the classical method referenced by Xu et al. [12]. The features obtained through this traditional extraction process were then introduced to the LSTM model for classification. Again, this adjustment was made while keeping all other experimental parameters unchanged.

LSTM model ablation: To examine the specific contribution of the LSTM model, we replaced it with a Random Forest algorithm for the classification task, employing parameter settings aligned with those suggested by Xu [12]. This substitution allowed us to directly compare the efficacy of LSTM with an alternative, well-established classification technique.

The outcomes of these ablation experiments, as depicted in Figure 8, exhibit significant variations when juxtaposed with the results achieved using our whole method. Such differences highlight each component's critical role in our methodology, validating their collective necessity and effectiveness in enhancing classification accuracy.

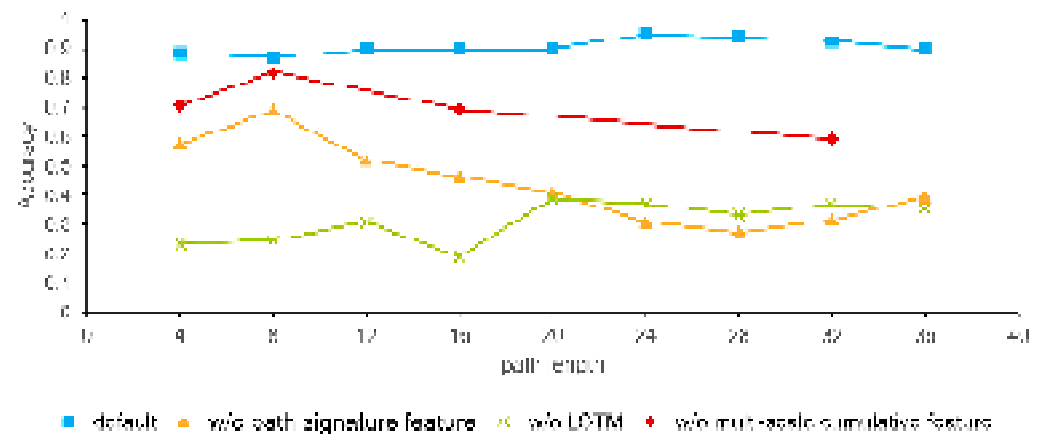


Figure 8. Results of ablation studies.

Through these ablation studies, the integral value and synergistic effect of the path signature features, the novel feature extraction method, and the LSTM model were demonstrated, affirming their indispensable contributions to our method's superior performance.

4.8. Comparison Experiments

4.8.1. The Benchmark Methods

To ascertain the effectiveness of our proposed encrypted traffic classification method, it was benchmarked against 11 contemporary mainstream approaches. These methodologies span across the following:

1. Fingerprint construction: FlowPrint [17];
2. Statistical methods: AppScanner [9], CUMUL [35], K-FP (K-Fingerprinting) [42], GRAIN [43], and ETC-PS [12];
3. Deep learning models: FS-net [20], DF [19], MT-FlowFormer [23], GraphDApp [44], and ET-BERT (flow) [16].

Notably, our proposed method exclusively utilizes flow-level features of encrypted traffic. For the sake of fairness, we opted to compare it against ET-BERT (flow), which also relies solely on flow-level features, rather than ET-BERT (packet), which achieves better classification performance but requires packet-level features.

4.8.2. Experimental Results

We evaluated the effectiveness of various methods, including our own, on the ISCX VPN-nonVPN and ISCX Tor-nonTor datasets, as shown in Tables 6 and 7. Table 6 presents the classification performance of traffic service types in VPN and non-VPN environments, while Table 7 shows the classification performance in Tor and non-Tor environments. Our method demonstrated a superior and competitive performance across most metrics compared to most benchmark methods.

Table 6. Classification performance of different methods for ISCX VPN-nonVPN dataset.

Dataset	ISCX-VPN				ISCX-nonVPN			
Method	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
FlowPrint [17]	0.8538	0.7451	0.7917	0.7566	0.6944	0.7073	0.7310	0.7131
AppScanner [9]	0.8889	0.8679	0.8851	0.8722	0.7576	0.7594	0.7465	0.7486
CUMUL [35]	0.7661	0.7531	0.7852	0.7644	0.6187	0.5941	0.5971	0.5897
K-FP [42]	0.8713	0.8750	0.8748	0.8747	0.7551	0.7478	0.7354	0.7387
GRAIN [44]	0.8129	0.8077	0.8109	0.8027	0.6667	0.6532	0.6664	0.6567
ETC-PS [12]	0.8889	0.8803	0.8937	0.8851	0.7273	0.7414	0.7133	0.7208
FS-net [20]	0.9298	0.9263	0.9211	0.9234	0.7626	0.7685	0.7534	0.7355
DF [19]	0.8012	0.7799	0.8152	0.7921	0.6742	0.6857	0.6717	0.6701
MT-FlowFormer [23]	0.9327	0.9152	0.9243	0.9193	0.8549	0.8473	0.8268	0.8344
GraphDApp [44]	0.6491	0.5668	0.6103	0.5740	0.4495	0.4230	0.3647	0.3614
ET-BERT (flow) [16]	0.9532	0.9436	0.9507	0.9463	0.9167	0.9245	0.9229	0.9235
Proposed	0.9474	0.9480	0.9474	0.9472	0.9053	0.9064	0.9053	0.9050

Table 7. Classification performance of different methods for ISCX Tor-nonTor dataset.

Dataset	ISCX-Tor				ISCX-nonTor			
Method	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
FlowPrint [17]	0.2400	0.0300	0.1250	0.0484	0.5243	0.7590	0.6074	0.6153
AppScanner [9]	0.7543	0.6629	0.6042	0.6163	0.9153	0.8435	0.814	0.8273
CUMUL [35]	0.6686	0.5349	0.4899	0.4997	0.8605	0.8143	0.7393	0.7627
K-FP [42]	0.7771	0.7417	0.6209	0.6313	0.8741	0.8653	0.7792	0.8167
GRAIN [44]	0.6914	0.5253	0.5346	0.5234	0.7895	0.6714	0.6615	0.6613
ETC-PS [12]	0.7486	0.6811	0.5929	0.6033	0.9155	0.8710	0.8311	0.8486
FS-net [20]	0.8286	0.7487	0.7197	0.7242	0.9278	0.8368	0.8254	0.8285
DF [19]	0.6514	0.4803	0.4767	0.4719	0.8568	0.8003	0.7415	0.7590
MT-FlowFormer [23]	0.8750	0.8252	0.8217	0.8220	0.8941	0.8742	0.8651	0.8670
GraphDApp [44]	0.4286	0.2557	0.2509	0.2281	0.6936	0.5447	0.5398	0.5352
ET-BERT(flow) [16]	0.9543	0.9242	0.9606	0.9397	0.9029	0.8560	0.8217	0.8332
Proposed	0.9386	0.9400	0.9386	0.9385	0.9503	0.9510	0.9503	0.9502

In these classification tasks, our method exhibited significant improvements in accuracy, precision, recall, and F1 score compared to ETC-PS, which also utilizes path signature features. When compared to the highest performing model, ET-BERT (flow), our method's accuracy was only 0.58% lower in the VPN environment and 1.57% lower in the Tor environment classification tasks. However, it is essential to acknowledge that ET-BERT (flow) is a complex, large-scale pre-trained model, starkly contrasting our more streamlined approach using a classical small-scale LSTM model.

4.8.3. Performance Analysis

To more comprehensively evaluate the model's complexity and the computational resources required, we used parameters (Params), floating-point operations (FLOPs), inference time, and latency time as evaluation metrics. Parameters denote the cumulative sum of all learnable weights and biases within a neural network, while FLOPs primarily refer to the number of floating-point arithmetic operations required during the neural network's forward and backward propagation processes.

In our experiment, we used a batch size of 32 samples. The time taken from inputting the model to obtaining the prediction results is referred to as the inference time, whereas latency time refers to the duration from loading the 32 samples into memory to obtaining the prediction results. It is important to note that the number of parameters and FLOPs can only be used to evaluate deep learning models. Therefore, we only assessed methods that utilize deep learning models.

Table 8 presents the Params, FLOPs, inference time, and latency time for different methods. The data in the table clearly show that our method exhibits significant advantages in several aspects for the same encrypted traffic classification task.

Table 8. The complexity of different methods.

Method	Params (M)	FLOPs (G)	Inference Time (ms)	Latency Time (ms)
FS-net [20]	2.17	24.86	39.94	45.72
DF [19]	1.83	3.06	29.57	33.56
MT-FlowFormer [23]	0.26	1.07	57.79	63.08
GraphDApp [44]	0.22	0.59	78.43	84.72
ET-BERT (flow) [16]	85.70	10.87	104.04	110.94
Proposed	0.03	7.32×10^{-4}	4.02	5.56

Firstly, our method uses significantly fewer parameters (Params) than other methods in terms of model complexity. This means that our model structure is more concise and efficient, which helps reduce the risk of overfitting while improving the model's interpretability and maintainability.

Secondly, regarding computational resource requirements, our method requires fewer floating-point operations (FLOPs). This indicates that our model demands less computation during forward and backward propagation, allowing faster training and inference processes. This is particularly important for resource-constrained applications, such as embedded systems or mobile devices.

Furthermore, our method demonstrates clear advantages in both inference time and latency time. The inference time is shorter, and the latency time is lower. This indicates that our model is not only computationally efficient but also highly responsive, making it suitable for real-time or near-real-time applications. Considering all these factors, we deem the slight discrepancy in classification accuracy to be acceptable.

We attribute the short inference time and latency time of our method to the following two main reasons:

- Our model architecture has a lower complexity and requires fewer computational resources. As a result, the calculations are completed quickly once the samples are input into the model, thereby reducing the inference time.
- Our sample size is relatively small, with an input size of (1,23,56). Additionally, the computational load on the CPU is minimal, allowing for a swift transfer of data from the CPU to the GPU, thereby reducing the latency time.

4.8.4. Statistical Analysis

To further validate the effectiveness of our method, we employed the Wilcoxon signed-rank test to determine if there are statistically significant differences between our classification results and those of other models. The Wilcoxon signed-rank test is a widely used statistical method for comparing two sets of data [45]. Our null hypothesis posits the following: "The two detection methods have the same classification performance". If the p -value is greater than 0.05, we accept the null hypothesis. If the p -value is less than 0.05, we reject the null hypothesis at a 95% confidence level. The p -values comparing our proposed method with each baseline model are shown in Tables 9 and 10.

Table 9. The p -values comparing our method with fingerprint construction and statistical methods.

Method	FlowPrint [17]	AppScanner [9]	CUMUL [35]	K-FP [42]	GRAIN [44]	ETC-PS [12]
p -value	0.006956	0.036267	0.000186	0.005447	0.000104	0.041655

Table 10. The p -values comparing our method with deep learning models.

Method	FS-Net [20]	DF [19]	MT-FlowFormer [23]	GraphDApp [44]	ET-BERT (Flow) [16]
p -value	0.015764	0.000183	0.016351	0.000322	0.167007

Based on the results, we can conclude that the p -values for all methods compared to our method, except for ET-BERT, are less than 0.05. This indicates that, except for ET-BERT, there are significant differences in classification performance between our method and the other methods. Specifically, a p -value less than 0.05 means we can reject the null hypothesis at a 95% confidence level, implying that the assumption that the two methods have the same classification performance does not hold.

Therefore, our method significantly outperforms the other baseline models in terms of classification performance, except for ET-BERT. For ET-BERT, the p -value is greater than 0.05, indicating no statistically significant difference in classification performance between our method and ET-BERT. This further validates the effectiveness of our method, demonstrating that the advantages of our model in complexity and computational resources do not come at the expense of classification performance.

5. Conclusions and Future Works

In this study, we proposed a method that combines path signature features with an LSTM model to classify encrypted traffic service types using only 24 consecutive inter-traffic features. The core of our method involves extracting fundamental features (such as packet size and arrival time) from the sequence of encrypted traffic packets to construct detailed traffic paths. These paths undergo a transformation process and are enhanced using multi-scale, cumulative feature extraction techniques, ultimately generating multi-scale path signature features. These features are then effectively used as inputs to the LSTM model, resulting in precise classification outcomes.

We evaluated our method on the publicly available ISCX VPN-nonVPN and ISCX Tor-nonTor datasets. The experimental results showed classification accuracies of 94.74%, 90.53%, 93.86%, and 95.03% in the VPN, non-VPN, Tor, and non-Tor environments, respectively. These results are comparable to the state-of-the-art methods. Additionally, our performance analysis and statistical analysis experiments demonstrated the advantages of our method over large-scale neural network models.

However, as a classification method that relies on packet length, our approach, like other methods [12] based on packet length features, is extremely limited in effectiveness against encrypted traffic that uses packet length obfuscation techniques such as packet padding [46]. Packet padding is an effective defense against traffic analysis, making all packet lengths identical. Exploring more distinguishable features in encrypted traffic may address this issue. Additionally, our model employs a relatively simple LSTM structure; it may thus perform poorly when handling more complex real-world classification tasks, indicating a need for more robust classifiers.

In future work, we plan to expand our approach by utilizing more datasets to apply it to more intricate network environments. Furthermore, the exploration of our methodology's applicability to the domain of anomaly traffic detection remains a central pillar of our future research objectives.

Author Contributions: Conceptualization, Y.M. and N.L.; methodology, Y.M.; software, Y.M.; validation, Y.M., G.Z. and X.Y.; formal analysis, Y.M.; investigation, G.Z.; resources, Y.M.; data curation, X.Y.; writing—original draft preparation, Y.M.; writing—review and editing, Y.M.; visualization, G.Z.; supervision, N.L.; project administration, N.L.; funding acquisition, N.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded in part by the National Social Science Fund of China under Grant 20&ZD293.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: <https://www.unb.ca/cic/datasets/vpn.html> (accessed on 1 June 2024).

Acknowledgments: The authors would like to thank the anonymous reviewers for their contribution to this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bader, O.; Lichy, A.; Hajaj, C.; Dubin, R.; Dvir, A. MalDIST: From Encrypted Traffic Classification to Malware Traffic Detection and Classification. In Proceedings of the 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2022; pp. 527–533.
2. Wang, Y.; He, H.; Lai, Y.; Liu, A.X. A Two-Phase Approach to Fast and Accurate Classification of Encrypted Traffic. *IEEE/ACM Trans. Netw.* **2023**, *31*, 1071–1086. [\[CrossRef\]](#)
3. Wang, W.; Zhu, M.; Wang, J.; Zeng, X.; Yang, Z. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), Beijing, China, 22–24 July 2017; pp. 43–48.
4. Shen, M.; Liu, Y.; Zhu, L.; Xu, K.; Du, X.; Guizani, N. Optimizing Feature Selection for Efficient Encrypted Traffic Classification: A Systematic Approach. *IEEE Netw.* **2020**, *34*, 20–27. [\[CrossRef\]](#)
5. Karagiannis, T.; Papagiannaki, K.; Faloutsos, M. BLINC: Multilevel traffic classification in the dark. *ACM SIGCOMM Comput. Commun. Rev.* **2005**, *35*, 229–240. [\[CrossRef\]](#)
6. Lotfollahi, M.; Jafari Siavoshani, M.; Shirali Hossein Zade, R.; Saberian, M. Deep packet: A novel approach for encrypted traffic classification using deep learning. *Soft Comput.* **2020**, *24*, 1999–2012. [\[CrossRef\]](#)
7. Al-Naami, K.; Chandra, S.; Mustafa, A.; Khan, L.; Thuraisingham, B.M. Adaptive encrypted traffic fingerprinting with bi-directional dependence. In Proceedings of the Conference on Computer Security Applications, Los Angeles, CA, USA, 5–9 December 2016.
8. Taylor, V.F.; Spolaor, R.; Conti, M.; Martinovic, I. Robust Smartphone App Identification via Encrypted Network Traffic Analysis. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 63–78. [\[CrossRef\]](#)
9. Taylor, V.F.; Spolaor, R.; Conti, M.; Martinovic, I. AppScanner: Automatic Fingerprinting of Smartphone Apps from Encrypted Network Traffic. In Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P), Saarbrücken, Germany, 21–24 March 2016; pp. 439–454.
10. Xie, G.; Li, Q.; Jiang, Y. Self-attentive deep learning method for online traffic classification and its interpretability. *Comput. Netw.* **2021**, *196*, 108267. [\[CrossRef\]](#)
11. Chen, K.T. Integration of Paths—A Faithful Representation of Paths by Noncommutative Formal Power Series. *Trans. Am. Math. Soc.* **1958**, *89*, 395–407. [\[CrossRef\]](#)
12. Xu, S.-J.; Geng, G.-G.; Jin, X.-B.; Liu, D.-J.; Weng, J. Seeing traffic paths: Encrypted traffic classification with path signature features. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2166–2181. [\[CrossRef\]](#)
13. Yamansavascular, B.; Guvensan, M.A.; Yavuz, A.G.; Karsligil, M.E. Application identification via network traffic classification. In Proceedings of the 2017 International Conference on Computing, Networking and Communications (ICNC), Silicon Valley, CA, USA, 26–29 January 2017; pp. 843–848.
14. Chen, R.T.Q.; Rubanova, Y.; Bettencourt, J.; Duvenaud, D.K. Neural ordinary differential equations. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 07366.
15. Dainotti, A.; Pescapé, A.; Claffy, K.C. Issues and Future Directions in Traffic Classification. *IEEE Netw.* **2012**, *26*, 35–40. [\[CrossRef\]](#)
16. Lin, X.; Xiong, G.; Gou, G.; Li, Z.; Shi, J.; Yu, J. Et-bert: A contextualized datagram representation with pre-training transformers for encrypted traffic classification. In Proceedings of the ACM Web Conference 2022, Lyon, France, 25–29 April 2022; pp. 633–642.
17. Van Ede, T.; Bortolameotti, R.; Continella, A.; Ren, J.; Dubois, D.J.; Lindorfer, M.; Choffnes, D.; van Steen, M.; Peter, A. Flowprint: Semi-supervised mobile-app fingerprinting on encrypted network traffic. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 23–26 February 2020.
18. Biju, V.G.; Prashanth, C. Friedman and Wilcoxon evaluations comparing SVM, bagging, boosting, K-NN and decision tree classifiers. *J. Appl. Comput. Sci. Methods* **2017**, *9*, 23–47. [\[CrossRef\]](#)
19. Sirinam, P.; Imani, M.; Juarez, M.; Wright, M. Deep fingerprinting: Undermining website fingerprinting defenses with deep learning. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 1928–1943.
20. Liu, C.; He, L.; Xiong, G.; Cao, Z.; Li, Z. Fs-net: A flow sequence network for encrypted traffic classification. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference On Computer Communications, Paris, France, 29 April–2 May 2019; pp. 1171–1179.
21. Lin, K.; Xu, X.; Gao, H. TSCRNN: A novel classification scheme of encrypted traffic based on flow spatiotemporal features for efficient management of IIoT. *Comput. Netw.* **2021**, *190*, 107974. [\[CrossRef\]](#)
22. Zhang, H.; Yu, L.; Xiao, X.; Li, Q.; Mercaldo, F.; Luo, X.; Liu, Q. TFE-GNN: A Temporal Fusion Encoder Using Graph Neural Networks for Fine-grained Encrypted Traffic Classification. In Proceedings of the ACM Web Conference 2023, Austin, TX, USA, 30 April–1 May 2023; pp. 2066–2075.
23. Zhao, R.; Deng, X.; Yan, Z.; Ma, J.; Xue, Z.; Wang, Y. Mt-flowformer: A semi-supervised flow transformer for encrypted traffic classification. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 2576–2584.
24. Graham, B. Sparse arrays of signatures for online character recognition. *arXiv* **2013**, arXiv:1308.0371.
25. Gyurkó, L.G.; Lyons, T.; Kontkowski, M.; Field, J. Extracting information from the signature of a financial data stream. *arXiv* **2013**, arXiv:1307.7244.

26. Perez Arribas, I.; Goodwin, G.M.; Geddes, J.R.; Lyons, T.; Saunders, K.E. A signature-based machine learning model for distinguishing bipolar disorder and borderline personality disorder. *Transl. Psychiatry* **2018**, *8*, 274. [CrossRef] [PubMed]
27. Hambly, B.; Lyons, T. Uniqueness for the signature of a path of bounded variation and the reduced path group. *Ann. Math.* **2010**, *171*, 109–167. [CrossRef]
28. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
29. Shen, S.; Han, T.; Pang, J. Car drag coefficient prediction using long–short term memory neural network and LASSO. *Measurement* **2024**, *225*, 113982. [CrossRef]
30. Yeganeh, A.; Ahmadi, F.; Wong, Y.J.; Shadman, A.; Barati, R.; Saeedi, R. Shallow vs. Deep Learning Models for Groundwater Level Prediction: A Multi-Piezometer Data Integration Approach. *Water Air Soil. Pollut.* **2024**, *235*, 441. [CrossRef]
31. Zhao, L.; Farhi, N.; Valero, Y.; Christoforou, Z. Long short-time memory neural networks for human driving behavior modelling. *Transp. Res. Procedia* **2023**, *72*, 2589–2596. [CrossRef]
32. Eswarsai. Exploring Different Types of LSTMs. Available online: <https://medium.com/analytics-vidhya/exploring-different-types-of-lstms-6109bcb037c4> (accessed on 28 July 2024).
33. Sun, Y.; Wang, X.; Tang, X. Deeply learned face representations are sparse, selective, and robust. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015, Boston, MA, USA, 7–12 June 2015; pp. 2892–2900.
34. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
35. Panchenko, A.; Lanze, F.; Pennekamp, J.; Engel, T.; Zinnen, A.; Henze, M.; Wehrle, K. Website Fingerprinting at Internet Scale. In Proceedings of the NDSS 2016, San Diego, CA, USA, 21–24 February 2016.
36. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
37. Wilson, D.L. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans. Syst. Man Cybern.* **1972**, *2*, 408–421. [CrossRef]
38. Batista, G.E.; Prati, R.C.; Monard, M.C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 20–29. [CrossRef]
39. Draper-Gil, G.; Lashkari, A.H.; Mamun, M.S.I.; Ghorbani, A.A. Characterization of encrypted and vpn traffic using time-related. In Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP), Rome, Italy, 19–21 February 2016; pp. 407–414.
40. Lashkari, A.H.; Gil, G.D.; Mamun, M.S.I.; Ghorbani, A.A. Characterization of tor traffic using time based features. In Proceedings of the International Conference on Information Systems Security and Privacy, Rome, Italy, 29–31 May 2017; pp. 253–262.
41. Shapira, T.; Shavitt, Y. FlowPic: A generic representation for encrypted traffic classification and applications identification. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 1218–1232. [CrossRef]
42. Hayes, J.; Danezis, G. k-fingerprinting: A robust scalable website fingerprinting technique. In Proceedings of the 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, USA, 10–12 August 2016; pp. 1187–1203.
43. Zaki, F.; Afifi, F.; Abd Razak, S.; Gani, A.; Anuar, N.B. GRAIN: Granular multi-label encrypted traffic classification using classifier chain. *Comput. Netw.* **2022**, *213*, 109084. [CrossRef]
44. Shen, M.; Zhang, J.; Zhu, L.; Xu, K.; Du, X. Accurate decentralized application identification via encrypted traffic analysis using graph neural networks. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 2367–2380. [CrossRef]
45. Sun, P.; Li, S.; Xie, J.; Xu, H.; Cheng, Z.; Yang, R. GPMT: Generating practical malicious traffic based on adversarial attacks with little prior knowledge. *Comput. Secur.* **2023**, *130*, 103257. [CrossRef]
46. Yu, S.; Zhao, G.; Dou, W.; James, S. Predicted packet padding for anonymous web browsing against traffic analysis attacks. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 1381–1393. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Network Intrusion Detection Based on Deep Belief Network Broad Equalization Learning System

Miaolei Deng ^{1,2,*} , Chuanchuan Sun ^{1,2} , Yupei Kan ^{1,2}, Haihang Xu ^{1,2}, Xin Zhou ^{1,2} and Shaojun Fan ^{1,2}

¹ College of Information Science and Engineering, Henan University of Technology, Zhengzhou 450001, China; scc@stu.haut.edu.cn (C.S.); 2022930951@stu.haut.edu.cn (Y.K.); 2022930927@stu.haut.edu.cn (H.X.); 2023930889@stu.haut.edu.cn (X.Z.); 2023930813@stu.haut.edu.cn (S.F.)

² Henan International Joint Laboratory of Grain Information Processing, Zhengzhou 450001, China

* Correspondence: dengmiaolei@haut.edu.cn

Abstract: Network intrusion detection systems are an important defense technology to guarantee information security and protect a network from attacks. In recent years, the broad learning system has attracted much attention and has been introduced into intrusion detection systems with some success. However, since the traditional broad learning system is a simple linear structure, when dealing with imbalanced datasets, it often ignores the feature learning of minority class samples, leading to a poorer recognition rate of minority class samples. Secondly, the high dimensionality and redundant features in intrusion detection datasets also seriously affect the training time and detection performance of the traditional broad learning system. To address the above problems, we propose a deep belief network broad equalization learning system. The model fully learns the large-scale high-dimensional dataset via a deep belief network and represents it as an optimal low-dimensional dataset, and then introduces the equalization loss v2 reweighing idea into the broad learning system and learns to classify the low-dimensional dataset via a broad equalization learning system. The model was experimentally tested using the CICIDS2017 dataset and fully validated using the CICIDS2018 dataset. Compared with other algorithms in the same field, the model shortens the training time and has a high detection rate and a low false alarm rate.



Citation: Deng, M.; Sun, C.; Kan, Y.; Xu, H.; Zhou, X.; Fan, S. Network Intrusion Detection Based on Deep Belief Network Broad Equalization Learning System. *Electronics* **2024**, *13*, 3014. <https://doi.org/10.3390/electronics13153014>

Academic Editor: Rajendra V. Boppana

Received: 17 June 2024

Revised: 26 July 2024

Accepted: 28 July 2024

Published: 31 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: network intrusion detection system; broad learning system; imbalanced datasets; equalization loss v2; CICIDS2017; CICIDS2018

1. Introduction

With the rapid development of modern computer technology and network information technology, networks affect every aspect of our lives, network attacks have become increasingly complex, and network security has never been more important. Although the Internet has brought people a lot of convenience, it also faces many new challenges. Network security problems are becoming increasingly serious, such as phishing websites, distributed denial-of-service attacks, and network worms threatening network security. Network intrusion detection systems (NIDSs) were proposed by Anderson [1] in 1980. The working principle of an intrusion detection system is mainly to monitor and filter network behaviors by analyzing host audit data, network traffic data, and other characteristics, identifying abnormal accesses in network communications, and notifying network administrators promptly, to achieve the purpose of ensuring network information security.

In recent years, due to the booming development of artificial intelligence, machine learning algorithms can better handle complex data for automated learning compared with traditional intrusion detection systems and have better adaptability and generalization. Therefore, more and more researchers have introduced machine learning technology into the intrusion detection system [2] and achieved successes [3]. However, traditional machine learning models perform poorly when dealing with complex and high-dimensional

data [4], and the robustness of the machine learning model constrains the performance of the model [5]. Deep learning can automatically extract and fully learn the complex features in the data to achieve high accuracy and robustness, and thus, deep learning techniques are widely used in intrusion detection systems [6]. However, the deep learning model training process is computationally resource-intensive, with long training times and high energy consumption [7]. In addition, deep learning models are usually black-box models that lack interpretability, making it difficult to understand and explain their internal decision-making process, which hampers the performance of intrusion detection systems. Moreover, the robustness of deep learning models also constrains the security of intrusion detection systems. Abbas et al. [8] conducted an in-depth study on the robustness of the models and achieved some success. More and more scholars [9] are beginning to adopt new techniques in the field of intrusion detection.

Recently, Chen et al. [10] proposed a new randomized neural network named the broad learning system (BLS), which is based on a flat network architecture. This model is different from the extend-to-depth deep learning structure in that it does not need to use gradient descent to update the weights, and the network structure is small in size and therefore computationally faster [11]. In addition, the BLS weight update is based on pseudo-inverse operations with better model interpretability and scalability. Therefore, some people have introduced BLS into intrusion detection systems with good results [12]. However, on the one hand, NIDS datasets tend to be severely imbalanced, and traditional BLS models cannot handle imbalanced datasets well, leading to very poor detection performance for minority classes of attacks, which is fatal for intrusion detection systems detecting attacks. On the other hand, NIDS datasets tend to have high dimensionality and contain more redundant information, which can lead to more time consumption and lower detection performance of the BLS model. Therefore, certain measures must be employed to address these issues to improve the detection performance of the model.

For the problem of dataset imbalance in intrusion detection, the traditional solution is to change the number of minority class samples or majority class samples by over-sampling or under-sampling to equalize the samples. For example, the synthetic minority over-sampling technique (SMOTE) [13] increases the number of minority class samples by synthesizing new minority class samples to balance the number of samples of different classes in the dataset. However, the samples generated by SMOTE may expand the original sample space and change the distribution of the sample data, leading to poor classification or even confusion of the model in the boundary region, and often causing more time cost due to the introduction of new samples. For the problem of high dimensionality of datasets in intrusion detection, the traditional method is based on the principal component analysis (PCA) dimensionality reduction [14]. Although it can reduce the dimensionality of the dataset, as a linear dimensionality reduction method, it can only find the linear correlation structure in the data, and for the non-linear structure of intrusion detection datasets, PCA cannot effectively capture this information, and its dimensionality reduction is achieved by projecting the data to a new low-dimensional space that often results in a serious loss of information in the original data.

To address the above problems, we propose a deep belief network broad equalization learning system (DBELS), in which the deep belief network (DBN) performs data dimensionality reduction on the NIDS dataset to eliminate redundant information. The equalization loss v2 (EQL v2) idea is introduced into the BLS, and the model balances the different classes of samples by adjusting the positive and negative gradient factors to enhance the performance in detecting attack samples of minority classes and classifies low-dimensional datasets to achieve the purpose of classifying and detecting attacks. The main contributions of this paper are as follows.

- We introduce the EQL v2 idea into the BLS and propose a new model by adding positive and negative gradient factors and recalculating its weights, which improves the poor learning ability of the BLS model for minority class samples by adjusting the positive and negative gradient factors and mitigates the defect of the BLS model in

that it is not good at dealing with the imbalanced dataset, to improve the performance of detecting minority class samples.

- We evaluated two types of DBN-based dimensionality reduction models—the traditional DBN model and the D_eBN model that introduces the idea of EQL v2—and experimentally compared the effects of the two dimensionality reduction models on the classification effect.
- We conducted many experiments on the DBELS model based on the publicly available benchmark dataset CICIDS2017, giving detailed experimental setups including binary and multi-classification, and evaluating the model in terms of accuracy, recall, false positive rate, time, and the receiver operating characteristic curves, finding obvious advantages over other models. We also tested the effect of hyperparameters on the model through many experiments and conducted comparative analyses with other state-of-the-art models.
- We further validated the fitness and scalability of the proposed model with the CICIDS2018 dataset. The performance and usefulness of the proposed model were evaluated by comparative analysis with other models.

The rest of this paper is organized as follows: Section 2 reviews the current research on intrusion detection systems based on the BLS, data dimensionality reduction, and data imbalance. In Section 3, we describe the details of the proposed DBELS. Detailed experimental results are reported and analyzed in Sections 4 and 5. Section 6 concludes this paper.

2. Related Works

This section focuses on the current state of research and the shortcomings of the broad learning system (BLS)-based network intrusion detection system (NIDS). We give traditional solutions to the limitations of the traditional BLS and analyze the advantages and disadvantages of these methods.

2.1. NIDS Based on BLS

The BLS model is a shallow network model that transforms and learns features by mapping nodes and enhancement nodes and calculates the model weights quickly by pseudo-inverse operations, which makes it simple and quick to train. When it was proposed, broad learning aroused widespread interest, and its applications are being investigated in many research fields, such as computer vision, image processing, medical data analysis, and natural language processing. There are also some works in the field of network intrusion detection. Li et al. [15] proposed a hybrid intrusion detection model based on the recurrent neural network (RNN) and BLS, and the experimental results show that the model based on the BLS has a better training effect and a shorter training time. Laura et al. [16] implemented the cascade of feature mapping nodes, a cascade of enhancement nodes, and a cascade of feature mapping nodes and enhancement nodes. The authors concluded that the cascade of enhancement nodes requires a longer training time than other BLS variants. Subsequently, they proposed a DDoS detection system based on broad learning for communication networks [17]. The authors concluded that the best detection performance can usually be achieved by using a cascaded BLS. Li et al. [18] proposed a tri-broad learning system (TBLS) based on the BLS model, which learns features from three dimensions of temporal granularity, data content, and spatial granularity of the dataset. Experimentally, it was proved that the TBLS model can achieve better detection performance by learning features from the three dimensions.

Although the BLS model has achieved some success in the field of intrusion detection, as a shallow neural network, it relies too much on the topology, number of samples, and class information of the training samples, and is unable to cope with imbalanced datasets. This leads to it paying too much attention to the majority class samples, and it almost ignores the learning of the minority class samples when learning classification, which leads to poor detection performance on imbalanced datasets, especially for the minority

class samples, which are worse or even undetectable. Second, high-dimensional datasets increase the complexity of BLS models and increase computational and storage costs. In addition, the higher the data dimensionality, the greater the influence of noise and redundant features, which may mask useful information and reduce the accuracy and stability of the BLS model. Therefore, certain techniques are needed to adapt the BLS to imbalanced and high-dimensional datasets for intrusion detection.

2.2. NIDS Based on Data Imbalance

Data imbalance refers to the fact that in the real world, different classes of data have different distributions, where certain types of data are significantly underrepresented, which has a serious negative impact on model classification. The datasets of intrusion detection are often highly imbalanced, which is fatal to the training and detection performance of the model; for example, in the case of CICIDS2017, the normal behavior of Benign accounts for 82.248%, and the abnormal behavior of Attack accounts for only 17.752%. Attack samples of DoS/DDoS account for 14.571%, Port Scan accounts for 2.607%, Brute Force accounts for 0.389%, Web Attack accounts for 0.096%, and Botnet accounts for 0.088%; thus, it is obvious that the normal class of samples is much more than the attack class of samples, which will cause the model to learn the majority-class samples too much. In response to the imbalance of intrusion detection datasets, some scholars have already conducted in-depth studies. Wu T et al. [13] addressed imbalanced data in network intrusion detection using k-means clustering and the synthetic minority over-sampling technique (SMOTE). They clustered data, identified minority-class clusters, and applied SMOTE to generate synthetic samples, demonstrating effectiveness in experiments. Ahmad T et al. [19] developed a hybrid model combining feature selection and pattern mining. They used rule-based analysis for dimensionality reduction, SMOTE for balancing data, and adaptive boosting (AdaBoost) for learning, effectively detecting minority-class samples. With deep learning advancements, Hao X et al. [20] used GANs to create synthetic datasets, enhancing classifier performance on minority classes through realistic data generation.

However, most of the methods used to solve the data imbalance problem in intrusion detection are based on generating data, whether it is generating minority-class samples by over-sampling or reducing majority-class samples by under-sampling, which balances the different samples by changing the number of the dataset; however, this changes the data distribution of the original data, which may affect the model's effect of learning to classify the dataset. In addition, the over-sampling method will increase the number of datasets, which will increase the extra computational cost of the model and cause extra time consumption. In recent years, Jin et al. [21] proposed equalization loss v2 (EQL v2), which is guided by positive and negative gradients and re-adjusts the weights to balance the learning of each type of task and enhance the learning of minority-class samples, and has achieved certain results. Since it does not change the distribution and number of the original samples and adjusts the weights so that the model learns each class of samples in a more balanced way, this is of great significance for solving the imbalance problem of intrusion detection systems, and introducing this idea into the BLS is conducive to enhancing the detection performance of the model in the face of imbalanced datasets.

2.3. NIDS Based on Data Dimensionality Reduction

Researchers have proposed some solutions to cope with the high-dimensionality problem in the field of intrusion detection. Zhang B et al. [14] proposed an intrusion detection method using an enhanced principal component analysis (PCA) combined with the Gaussian plain Bayesian algorithm. By weighting the primary feature vectors in PCA, the method reduces data contamination. This approach, followed by the Bayesian algorithm for detection, significantly decreases detection time compared to using classifiers alone. Shen Z et al. [22] introduced an enhanced naive bayes classification algorithm that integrates principal component analysis with linear discriminant analysis to reduce sample space dimensionality and refines Bayesian computation by incorporating attribute

correlation. This approach allows the classifier to account for both attribute frequency and correlation issues, addressing real-time and accuracy challenges in intrusion detection with numerous features. Salo F et al. [23] developed a feature-processing integration technique that combines information gain and principal component analysis to extract a low-dimensional optimal subspace. They then merge multiple classifiers using support vector machines, instance learning algorithms, and multilayer perceptron decision strategies, employing a probability mean combination rule for voting. Testing across various datasets demonstrated that this method not only achieves high accuracy but also significantly reduces computational costs, making it more effective for large-scale data detection.

However, most of the current solutions to the problem of high dimensionality in NIDS datasets are machine learning algorithms, which reduce the dimensionality of the dataset to a certain extent, but still have greater limitations in the face of non-linear, high-dimensionality features, resulting in a lower detection rate of the model. Since deep learning algorithms are effective in learning high dimensionality, and non-linear features can autonomously complete the learning and feature extraction of raw data without too much human intervention, deep learning algorithms have been introduced into large-scale intrusion detection systems, which can better complete the extraction and dimensionality reduction of data features. The deep belief network (DBN), as a representative deep learning model, was proposed by Geoffrey Hinton et al. [24] in 2006. From the raw data, it can learn the multi-layered abstract feature representations that effectively capture the complex structure and changing patterns of the data. This hierarchical feature extraction process helps to map high-dimensional input data to a low-dimensional representation space while preserving the important information of the data for dimensionality reduction. Compared with traditional dimensionality reduction methods, DBN dimensionality reduction not only better maintains the structure and relevance of the data but also adaptively learns the non-linear relationships in the data, which improves the expressive ability and classification performance of the reduced data. Therefore, the DBN is superior in the task of dimensionality reduction for intrusion detection models.

3. Methodology

This section constructs the deep belief network broad equalization learning system (DBELS) based on BLS, EQL v2, and DBN in detail in three stages: data preprocessing, data dimensionality reduction, and model classification.

3.1. DBELS Architecture

The framework of DBELS is shown in Figure 1, which is divided into three stages. The first stage is data preprocessing, which mainly processes the original data into a data form suitable for the dimensionality reduction of the DBN. The CICIDS2017 dataset contains 79 features containing many redundant features that hamper the model training and classification. In this paper, data preprocessing includes two parts: data cleaning and data standardization. The purpose of data cleaning is to reduce redundant features by calculating the contribution of features to the classification through correlation coefficients, eliminate the bad values of the data, and avoid its negative interference effect on the model. The purpose of data standardization is to make the model focus on different features of the data in a balanced way, while ignoring the influence of features with different weights on the model due to the size of the value. Due to the serious data imbalance in the CICIDS2017 dataset, the over-sampling method SMOTE can be used to increase the minority attack samples when the DBN reduces dimensionality to alleviate the problem of too few samples for the minority class.

The second stage is DBN dimensionality reduction, which aims to represent the original high-dimensional data as optimal low-dimensional data. The DBN dimensionality reduction model we use adopts a classic structure, which maximizes the retention of core features while reducing the feature dimensions as much as possible. The model fully trains each restricted Boltzmann machine (RBM) through pre-training, adjusts the whole DBN

structure through weight fine-tuning, and fully learns the features of the standardized dataset, which is represented as an optimal low-dimensional dataset. The third stage uses the broad equalization learning system (BELS) to classify the optimal low-dimensional data. The BELS model equalizes the model's learning of the minority-class samples by adjusting the positive and negative gradient factors and increasing its weights to enhance the detection and recognition class of the minority-class samples.

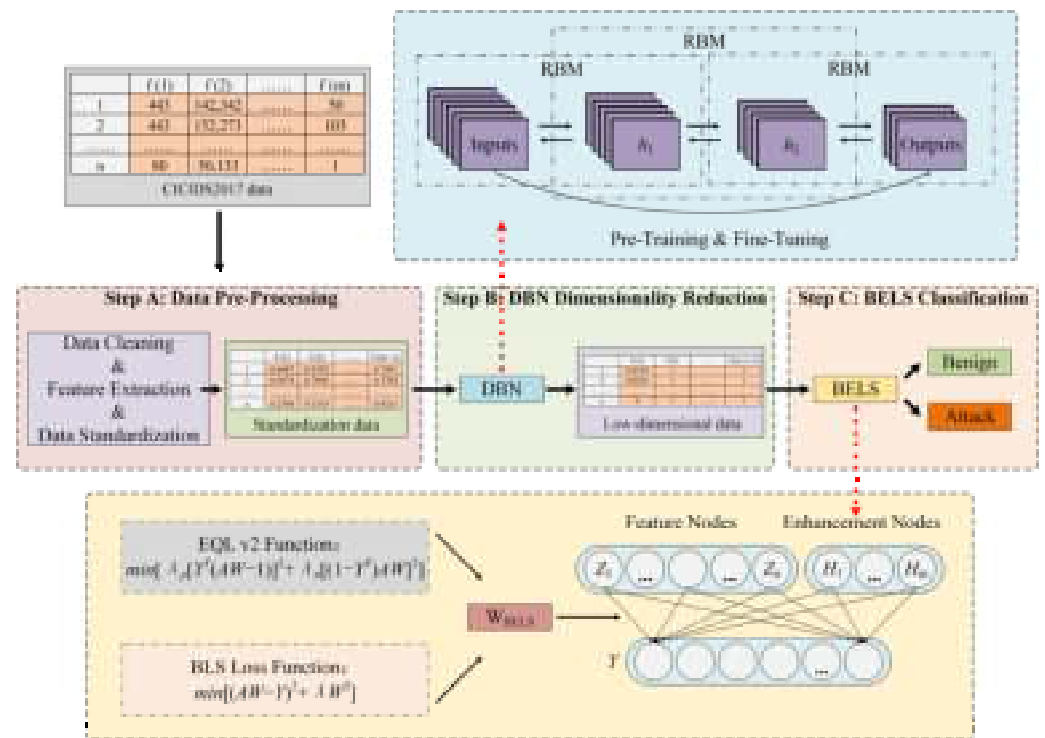


Figure 1. The structure of DBELS.

3.2. Deep Belief Network

3.2.1. Restricted Boltzmann Machine

The restricted Boltzmann machine [24] is the basic structure of the DBN, which is a shallow neural network and consists of a two-layer structure of visible and hidden layers. Connections in an RBM are undirected, and there are no connections between neurons in the same layer; its structure can be seen in Figure 2. Commonly used RBMs are generally binary. Whether they are in the hidden layer or the visible layer, their neuron takes the value of 0 or 1 only. The working principle of an RBM is based on the energy function, and the energy function of the visible layer and hidden layer are given by

$$E(v, h) = -\sum_i b_i v_i - \sum_j c_j h_j - \sum_{i,j} v_i h_j w_{ij} \quad (1)$$

where i belongs to the visible layer and j belongs to the hidden layer, b and c are model biases, w is weight of the model, the first part represents the contribution of the nodes of the visible layer to the energy of the system, the second part represents the contribution of the nodes of the hidden layer to the energy of the system, and the last part represents the contribution of the energy of the system due to the interaction between the visible layer and the hidden layer.

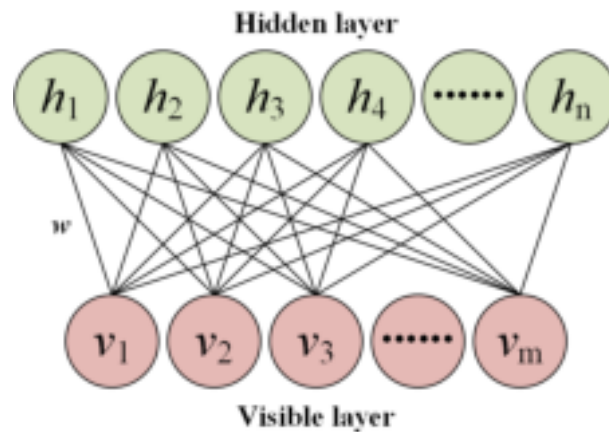


Figure 2. The structure of RBM.

The RBM defines the joint probability density of the visible and hidden layers in terms of an energy function. The joint probability of the visible layer and hidden layer is given as

$$p(v, h) = \frac{1}{Z} e^{-E(v, h)} \quad (2)$$

where Z is the partition function, which is used to normalize the probability distribution and ensure that the sum of the probability distributions is equal to 1:

$$Z = \sum_{v, h} e^{-E(v, h)} \quad (3)$$

With the joint probability density of the visible and hidden layers, summing over h yields the marginal density distribution of the visible layer:

$$p(v) = \frac{1}{Z} \sum_h e^{-E(v, h)} \quad (4)$$

During the RBM model training process, the hidden layer variable h_1 is sampled and computed using the posterior probability $p(h_1|v)$, and the new visible layer variable v_1 is sampled by the posterior probability $p(v_1|h_1)$; subsequently, the new hidden layer variable h_2 is sampled again in a repetitive manner to obtain the visible layer v_2 . The above steps are repeated several times until the parameters converge or reach the predefined number of iterations, the joint probability distribution is close to the smooth distribution, and the RBM training is completed.

3.2.2. Training of Deep Belief Network

The deep belief network [24] is a deep neural network stacked by multiple RBMs. The DBN structure is shown in Figure 3. During the training process, the DBN first uses an unsupervised greedy layer-by-layer training algorithm for each layer of the RBM, except for the first and last layers of the DBN structure, where each layer of the RBM hidden layer serves as the visible layer of the next layer of the RBM, to transfer feature information and learning information (this phase is also called the pre-training phase of DBN). After the pre-training is completed, a small amount of labeled data are attached to the last layer of the DBN model, and supervised training is performed on each layer of the RBM. The backward propagation algorithm propagates the error information to each layer of the RBM and uses the maximum likelihood function for the objective function, which can optimize the whole DBN, thus obtaining the optimal low-dimensional representation of the original data.

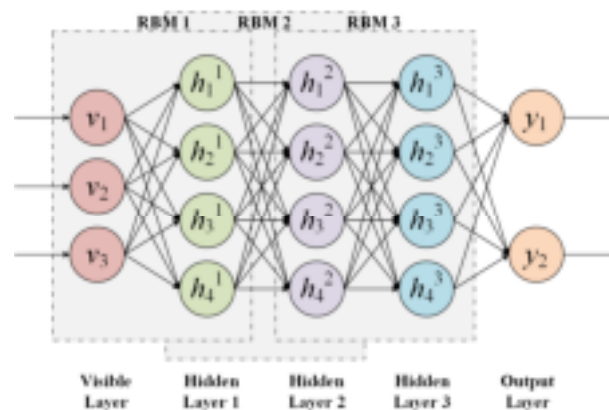


Figure 3. The structure of DBN.

3.3. Broad Learning System

The carrier of a broad learning system [10] is a random vector function linked neural network. Compared with the deep learning method, the BLS has only two layers of neural network, and its structure is in the form of width extension. Its model structure is shown in Figure 4. Firstly, the BLS maps the input data into a feature node matrix through the mapping function. Secondly, the feature node matrix forms an enhancement node matrix by using the enhancement function. Then, the feature mapping nodes and the enhancement nodes are used together as the input of the hidden layer. Finally, the pseudo-inverse operation is used to calculate the weight matrix between the hidden layer and the output layer to achieve the purpose of training the BLS model.

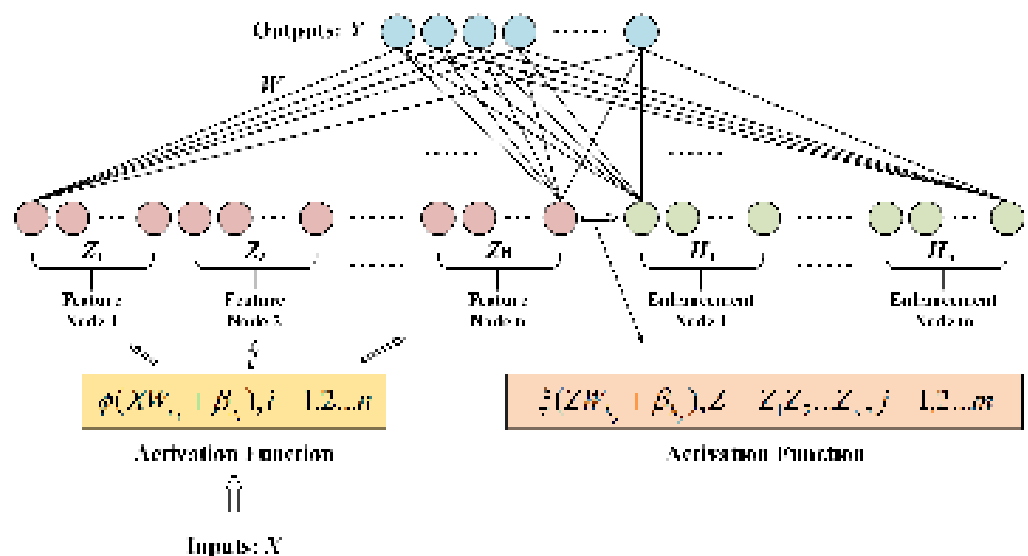


Figure 4. The structure of BLS.

Suppose that the input data $X = [x_1, x_2, \dots, x_n]$ belong to $R(N * M)$ and the output matrix, $Y = [y_1, y_2, \dots, y_n]$, belongs to $R(N * C)$, where N is the total data sample, M is the sample dimension, C is the number of categories of the data sample, $Z_1, Z_2, \dots, Z_n$ is the feature node matrix, $H_1, H_2, \dots, H_n$ is the enhancement node matrix, and W is the output weight matrix between the hidden layer and the output layer.

The specific procedure of the BLS is as follows. First, the sample data X are mapped by group i features to obtain the group i feature node matrix Z_i , where $i \in (1, 2, \dots, n)$:

$$Z_i = \phi_i(XW_{e_i} + \beta_{e_i}) \in \mathbb{R}^{N \times q_i} \quad i = 1, 2, \dots, n \quad (5)$$

where φ_i is the activation function, q is the number of feature nodes corresponding to each feature mapping, W_{e_i} is the weight matrix of the i th feature mapping, and β_{e_i} is the bias matrix of the i th feature mapping. Both W_{e_i} and β_{e_i} are randomly generated. Then, all feature node matrices are integrated to obtain the total feature node matrix:

$$Z^n = [Z_1, Z_2, \dots, Z_n] \in \mathbb{R}^{N \times nq} \quad (6)$$

Second, the total feature node matrix Z^n undergoes group j enhancement mapping to obtain the group j enhancement node matrix H_j , where $j \in (1, 2, \dots, m)$, is as follows:

$$H_j = \zeta_j(Z^n W_{h_j} + \beta_{h_j}) \in \mathbb{R}^{N \times r} \quad j = 1, 2, \dots, m \quad (7)$$

where ζ_j is the activation function, r represents the number of enhancement nodes corresponding to each group of enhancement transformations, W_{h_j} is the group j enhancement mapping weight matrix, and β_{h_j} is the group j enhancement mapping bias matrix. The total enhancement node matrix is obtained by integrating all enhancement node matrices:

$$H^m = [H_1, H_2, \dots, H_m] \in \mathbb{R}^{N \times mr} \quad (8)$$

Then, the feature mapping nodes are merged with the enhancement nodes as inputs to the BLS, defined as A :

$$A = [Z^n | H^m] \in \mathbb{R}^{N \times (nq + mr)} \quad (9)$$

Finally, the predicted value of the broad learning algorithm can be given as

$$\hat{Y} = [Z^n | H^m]W = AW \quad (10)$$

where W is the weight matrix of feature nodes and enhancement nodes to the output layer.

To minimize the error between the predicted value $\hat{Y}$ and the true value Y and to find a suitable W , the BLS model is optimized with the following function:

$$\arg \min_W F_{BLS} = \|Y - \hat{Y}\|_2^2 + \lambda \|W\|_2^2 = \|Y - AW\|_2^2 + \lambda \|W^m\|_2^2 \quad (11)$$

where the first term is used to control the minimization of the training error, the second term is used to prevent the model from overfitting, and the value λ denotes the further constraints on the sum of the squared weights; we take derivative of the above function and let it take the value of zero, obtaining W as

$$W = (A^T A + \lambda I)^{-1} A^T Y \quad (12)$$

where A is the mapping and enhancement matrix, λ is a penalty factor, I is an identity matrix, and Y is the true label.

3.4. Equalization Loss v2

The basic idea of equalization loss v2 [21] is to equalize the degree of learning of the model for different classes of samples by weighting the positive and negative gradients of each classifier according to the cumulative gradient ratios of the positive and negative gradients of the classifier, respectively. In solving the problem of imbalanced datasets, a classic strategy is the focal strategy, and Eqlv2 is improved on this basis. However, EQL v2 offers distinct advantages over focal loss, particularly in handling class imbalance without introducing bias towards minority classes. Unlike focal loss, which adjusts the loss contribution based on prediction difficulty, EQL v2 dynamically balances gradients during training, reducing overfitting on rare categories while maintaining focus on common classes. This approach enhances model stability and performance across varied datasets,

providing a more nuanced response to imbalance challenges. The positive and negative gradients for the output Z of each classifier concerning the loss $\mathcal{L}$ are

$$\nabla_{z_j}^{pos}(\mathcal{L}) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} y_j^i (p_j^i - 1) \quad (13)$$

$$\nabla_{z_j}^{neg}(\mathcal{L}) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} (1 - y_j^i) p_j^i \quad (14)$$

where $\mathcal{I}$ is the total number of samples, p_j^i is the probability that the i th sample is predicted to be of class j , and y_j^i is the value of the One-Hot truthful labeling matrix in which the i th sample is of class j .

EQL v2 weights are updated by first defining g_j as the ratio of the cumulative positive gradient to the negative gradient of task j up to iteration t . In this iteration, the weights of the positive gradient q_t and the negative gradient r_t can be computed as follows:

$$q_j^{(t)} = 1 + \alpha (1 - f(g_j^{(t)})) \quad (15)$$

$$r_j^{(t)} = f(g_j^{(t)}) \quad (16)$$

where f is a mapping function as

$$f(x) = \frac{1}{1 + e^{-\gamma(x-\mu)}} \quad (17)$$

The positive gradient $q_j^{(t)}$ and negative gradient $r_j^{(t)}$ are obtained and then they are applied to the positive and negative gradients of the current batch, respectively, and the reweighted gradient is

$$\nabla_{z_j}^{pos'}(\mathcal{L}^{(t)}) = q_j^{(t)} \nabla_{z_j}^{pos}(\mathcal{L}^{(t)}) \quad (18)$$

$$\nabla_{z_j}^{neg'}(\mathcal{L}^{(t)}) = r_j^{(t)} \nabla_{z_j}^{neg}(\mathcal{L}^{(t)}) \quad (19)$$

Finally, the ratio g_j^{t+1} of the accumulated positive gradient to the negative gradient of the iteration $t + 1$ update is computed and denoted as

$$g_j^{t+1} = \frac{\sum_{t=0}^T |\nabla_{z_j}^{pos'}(\mathcal{L}^{(t)})|}{\sum_{t=0}^T |\nabla_{z_j}^{neg'}(\mathcal{L}^{(t)})|} \quad (20)$$

EQL v2 chooses the gradient statistic as a measure of whether a task is in a balanced training state, balancing the model based on how easy or difficult each classification is for the model to train, rather than simply considering the number of its positive and negative samples. Introducing the idea of gradient-guided weighting to DBN and BLS helps the model to be trained with more attention to the features of minority classes of samples, thus making the model more comprehensive and balanced in evaluating each class of samples. The model adjusts the optimal weight contribution of the iteration based on the impact of different categories of samples on the performance of the classification.

3.5. Broad Equalization Learning System

The W of the broad equalization learning system is solved as follows; based on the EQL v2 positive and negative gradients, the sample positive and negative gradients can be given as follows:

$$p(W) = Y^T(AW - 1) \quad (21)$$

$$n(W) = (1 - Y^T)AW \quad (22)$$

where A is the feature matrix of the mapped and enhancement nodes, Y is the true label, and W is the weight matrix.

The gradient function can be constructed from the positive and negative gradients as

$$G(W) = \lambda_p [Y^T (AW - 1)]^2 + \lambda_n [(1 - Y^T)AW]^2 \quad (23)$$

where λ_p is the positive gradient factor and λ_n is the negative gradient factor. The smaller $G(W)$ is, the greater the probability of positive samples, and the smaller the probability of negative samples. Adjusting the parameters λ_p and λ_n can adjust the learning effect of the model on different samples to achieve equalization of the degree of contribution of each class of samples to the model learning classification.

Based on the BLS loss function,

$$L(W) = (AW - Y)^2 + \lambda W^2 \quad (24)$$

where λ is the penalty factor; then, the optimization objective of the BELS is the EQL v2 gradient function and the BLS loss function:

$$\operatorname{argmin} : F(W) = (AW - Y)^2 + \lambda W^2 + \lambda_p [Y^T (AW - 1)]^2 + \lambda_n [(1 - Y^T)AW]^2 \quad (25)$$

Among them, the first part is the loss function, which is used to control the classification error of the model. The second part is the penalty term, which mainly prevents the model from overfitting. The third part is the gradient term, which is used to balance the learning degree of different samples and alleviate the poor performance of the model in detecting samples of minority classes.

The existing $F(W)$ is expressed as $\frac{\partial F(W)}{\partial W}$ by taking the partial derivative of W :

$$W = [\lambda_p A^T Y Y^T A + \lambda_n A^T (J_1 - Y)(J_2 - Y^T)A + A^T A + \lambda I]^{-1} (\lambda_p A^T Y Y^T J_3 + A^T Y) \quad (26)$$

where J is a matrix with all elements equal to 1, λ_p is a positive gradient factor, λ_n is a negative gradient factor, the positive and negative gradient factors are used to control the learning degree of the model on different samples, and λ is a penalty factor to avoid the overfitting of the model caused by the excessively large W .

4. Experiments

This section presents the detailed experiments. We analyze in detail the effect of hyperparameters on the model. We employ grid search to fix the positive and negative gradient factors within appropriate ranges, seeking the optimal parameter combination that maximizes the detection performance of the model. To evaluate the effectiveness of the proposed method, we conduct complete ablation experiments. Finally, we give a comparative analysis of the proposed model with other state-of-the-art methods.

4.1. CICIDS2017 Datasets

The CICIDS2017 dataset [25] proposed by the Canadian institute for cybersecurity is one of the important benchmark datasets for evaluating intrusion detection models. The authors used the behavior profile system to analyze the abstract behavior of human interactions based on different protocols, constructing 25 abstract behaviors of users and generating natural friendly background traffic. The dataset contains benign and new common attacks and includes seven types of network attacks, namely denial-of-service attacks, secure shell brute force attacks, botnet attacks, distributed denial-of-service attacks, web application attacks, heartbleed exploits, and penetration testing attacks. The sample distribution and features of the dataset are shown in Table 1.

Table 1. Class distribution of CICIDS2017 dataset.

Category	Label	Sample	Feature
Benign	Benign	2,273,097	Dest Port, Flow Duration, Tot Fwd Bwd Pkts, Tot Len Fwd Bwd Pkts, Fwd Bwd Pkt Len
DoS/DDoS	DoS Hulk, DDoS, DoS GoldenEye, DoS Slowloris, DoS Slowhttptest, Heartbleed	380,699	Max Min Mean Std, Min Max Pkt Len, Pkt Len Mean Std Var, Avg Pkt Size, Avg Fwd Bwd Seg Size, Flow Bytes Pkts, Fwd Bwd Pkts, Fwd Bwd Avg Bulk Rate, Fwd Bwd Avg Bytes Pkts, Flow IAT
PortScan	PortScan	158,930	Mean Std Max Min, Fwd Bwd IAT
Brute Force	FTP-Patator, SSH-Patator	13,835	Total Mean Std Max Min, Down-Up Ratio, Active Idle Mean Std Max Min, Fwd Bwd PSH
Web Attack	Web Attack—Brute Force, Web Attack—XSS, Web Attack—SQL Injection	2180	Flags, Fwd Bwd URG Flags, Init Win Bytes
Botnet	Bot	1966	Fwd Bwd, Fwd Bwd Header Len, Act Data Pkt Fwd, FIN SYN RST PSH ACK URG CWE ECE
Total	-	2,830,707	Flags, Min Seg Size Fwd, Subflow Fwd Bwd Pkts Bytes, Label
			79

4.2. Implementation Details

The DBELS model parameters were set as in Table 2, including the training parameters of the DBN dimensionality reduction model, the BELS activation function, the number of nodes in each of the mapping groups, the number of nodes in each of the enhancement groups, and the penalty coefficient, λ . The DBN dimensionality reduction model consisted of a stack of three RBMs, (49,64), (64,32), and (32,16). All the experiments were conducted using a 64-bit Intel(R) Core (TM) i7-11700 CPU with 32 GB RAM in the Windows 11 environment. The models were implemented in Python v3.9.16 using the PyTorch v2.1.0 library.

Table 2. The parameters of DBELS.

Parameter	Pre-Training	Fine-Tuning	Description
Epochs	30	100	-
Learning rate	0.0001	0.00001	-
Batch size	64	128	-
Optimiser	SGD	Adam	-
Gibbs step	-	-	5
Mapping Enhancement group node count	-	-	16
Mapping Enhancement activation function	-	-	Relu
λ	-	-	0.001

4.3. Performance Metrics

The evaluation metrics for intrusion detection mainly include accuracy, recall, and false positive rate (*FPR*). In the formula, the true positive (*TP*) represents the number of samples correctly identified as positive, the true negative (*TN*) is the number of samples correctly identified as negative, the false positive (*FP*) is the number of negative samples incorrectly identified as positive, and the false negative (*FN*) is the number of positive samples incorrectly identified as negative. The *accuracy* denotes the ratio of the number of samples correctly classified by the model to the total number of samples:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (27)$$

The *recall* denotes the proportion of positive cases in the sample that are correctly predicted, also known as the detection rate:

$$Recall = \frac{TP}{TP + FN} \quad (28)$$

The false positive rate is the proportion of truly negative samples that the model incorrectly classifies as positive:

$$FPR = \frac{FP}{FP + TN} \quad (29)$$

4.4. Analysis of Hyperparameters

4.4.1. Effect of λ_p and λ_n on Binary Classification

For the two parameters λ_p and λ_n in the DBELS model, their effects on the performance of the model in different classification tasks were explored. Their effects on the performance of accuracy, recall, and time in the binary classification task are shown in Figure 5. For accuracy and recall, the performance tends to increase with decreasing λ_p and λ_n . Different parameter combinations lead to larger differences because the model is more sensitive to the two parameters in the binary classification task, and as the order of magnitude improves, the performance of the model changes drastically. The experiments show that when λ_p is around 10^{-5} orders of magnitude and λ_n is around 10^{-4} orders of magnitude, the model can maintain higher accuracy and recall, at around 0.99 and 0.98, respectively. For the training time of the model with the change in λ_p and λ_n , the training time fluctuates slightly above and below 1.6 s, indicating that the model training time in the binary classification task is not sensitive to the positive and negative factors.

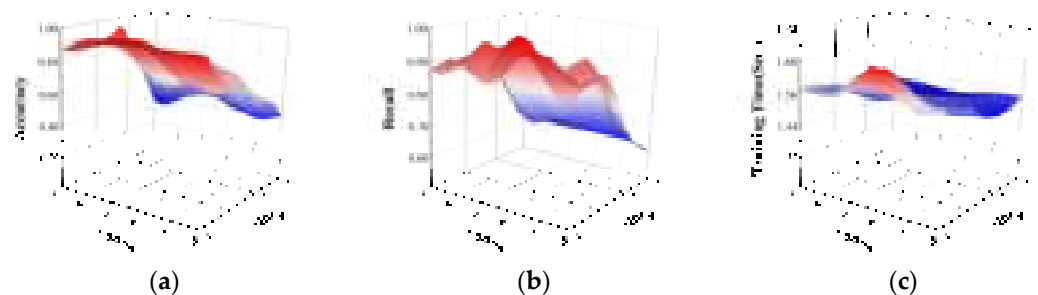


Figure 5. The effect of λ_p and λ_n on binary classification. The effects on the performance: (a) the effect on accuracy; (b) the effect on recall; (c) the effect on training time.

4.4.2. Effect of λ_p and λ_n on Multi-Classification

In the multi-classification task, its effect on the accuracy, recall, and time performance are shown in Figure 6. For accuracy, the model is almost insensitive to λ_n and almost tends to be stable as λ_n changes; when $10^{-3} < \lambda_p < 10^{-2}$, the accuracy increases as λ_p increases. For recall, as λ_p and λ_n increase, the recall of the model tends to increase. When the λ_p value's order of magnitude is 10^{-2} and the λ_n value's order of magnitude is about 10^{-3} , recall grows to a maximum of about 0.95. For the model training time with the change in λ_p and λ_n , the time of the model fluctuates from 1.4 s to 1.6 s and almost stabilizes, which shows that the training time is not sensitive to the positive and negative factor parameters. To achieve a better overall performance, the order of magnitude of λ_p is around 10^{-4} , and the order of magnitude of λ_n is around 10^{-3} , which can make the accuracy stabilize to 0.96 and the recall stabilize to 0.95.

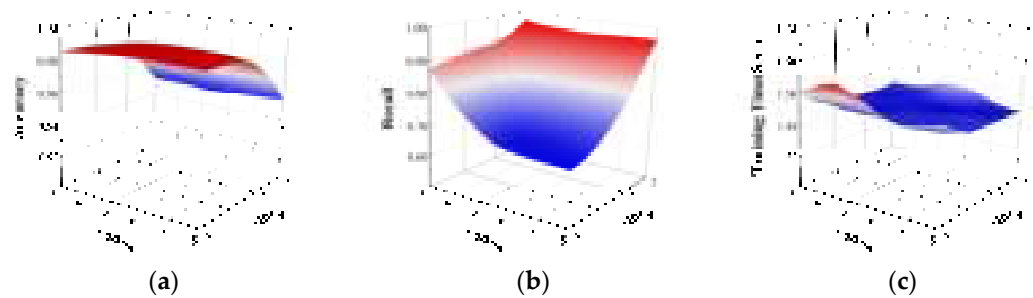


Figure 6. The effect of λ_p and λ_n on multi-classification. The effects on the performance: (a) the effect on accuracy; (b) the effect on recall; (c) the effect on training time.

4.4.3. Effect of Mapping and Enhancement Groups on Binary Classification

The experiments explored the effects of DBELS model enhancement groups and mapping groups on the performance of the model for different classification tasks. For the binary classification task, the effects of enhancement groups and mapping groups on accuracy, recall, and training time are shown in Figure 7. For accuracy and recall, the model floats more with the change in parameters because the binary classification task model is more sensitive to these two parameters. With the increase in the mapping matrix Z and the enhancement matrix H , its input matrix A increases, and the model detection performance is highly dependent on the matrix, A . For training time, it increases with the increase in enhancement groups and mapping groups, because the input matrix A increases with the increase in parameters. From the experimental results, it is known that when enhancement groups and mapping groups are selected as 1, it can ensure both less training time and higher accuracy and recall.

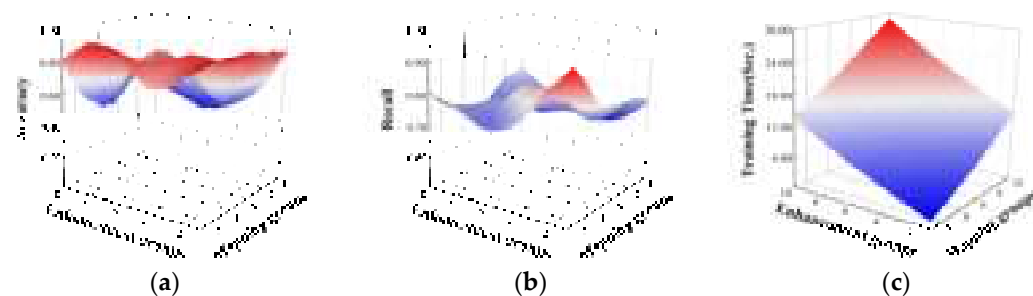


Figure 7. The effect of enhancement groups and mapping groups on binary classification. The effects on the performance: (a) the effect on accuracy; (b) the effect on recall; (c) the effect on training time.

4.4.4. Effect of Mapping and Enhancement Groups on Multi-Classification

For the multi-classification task, the effects on accuracy and recall are shown in Figure 8. With the increase in enhancement groups and mapping groups, the accuracy and recall almost tend to stabilize, indicating that for the multi-classification task, these two metrics of the DBELS model are not sensitive to the mapping and enhancement groups. The training time shows an upward trend in time consumption both with the increase in mapping groups and enhancement groups. The reason is that the increase in mapping groups and enhancement groups increases the size of the mapping matrix Z and the enhancement matrix H , resulting in an increase in matrix A , increasing the burden on and consumption of the model. It can be concluded that since the increase in enhancement groups and mapping groups has less impact on the detection performance of the model and increases the consumption of time cost when enhancement groups and mapping groups are selected as 1, the accuracy and recall maintain a higher performance and the time consumption is minimized.

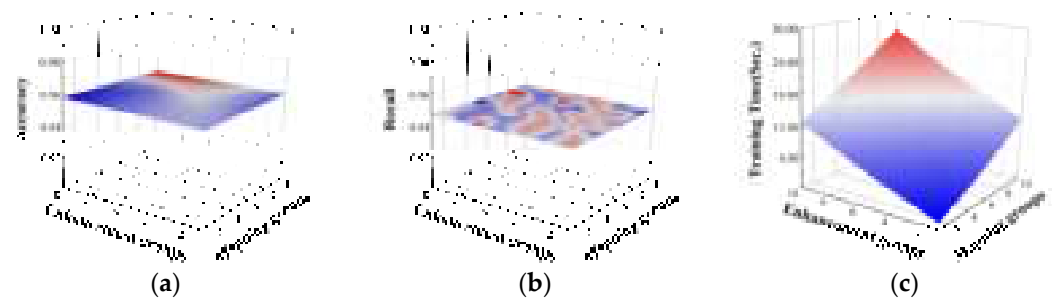


Figure 8. The effect of enhancement groups and mapping groups on multi-classification. The effects on the performance: (a) the effect on accuracy; (b) the effect on recall; (c) the effect on training time.

4.5. Ablation Studies

To evaluate the effectiveness of the proposed method, we performed ablation experiments based on three dimensionality reduction models and two classification models. The three dimensionality reduction models are traditional PCA, traditional DBN, and D_eBN with the introduction of EQL v2. The classification methods are traditional BLS and our improved BELS, respectively. We evaluated the detection performance of a total of eight methods under a binary classification task and multi-classification task based on the CICIDS2017 dataset. All the experiments were conducted in the same environment with identical forms of data processing, and the detection performances of different models are shown in Tables 3 and 4. The recognition rate of each sample under different classification tasks is shown in Tables 5 and 6.

Table 3. The performance of different methods on binary classification.

Method	Algorithm					Evaluation Metric		
	BLS	BELS	PCA	DBN	D _e BN	Accuracy	Recall	Time(s)
BLS	✓					0.88105	0.63062	1.83126
PBLS	✓		✓			0.86780	0.59044	0.75575
DBLS	✓			✓		0.91531	0.73711	0.60949
D _e BLS	✓				✓	0.29939	0.58103	0.60531
BELS		✓				0.96670	0.92483	4.86849
PBELS		✓	✓			0.92279	0.85250	1.64884
DBELS		✓		✓		0.99240	0.98579	1.63914
D _e BELS		✓			✓	0.98896	0.97516	1.57186

The ✓ in the table indicates that the method contains the algorithm.

Table 4. The performance of different methods on multi-classification.

Method	Algorithm					Evaluation Metric		
	BLS	BELS	PCA	DBN	D _e BN	Accuracy	Recall	Time(s)
BLS	✓					0.97450	0.47382	1.74388
PBLS	✓		✓			0.93755	0.29756	0.51006
DBLS	✓			✓		0.99286	0.65797	0.51341
D _e BLS	✓				✓	0.98936	0.67712	0.50913
BELS		✓				0.97060	0.52527	5.00825
PBELS		✓	✓			0.93848	0.47024	1.77999
DBELS		✓		✓		0.98719	0.80346	1.49983
D _e BELS		✓			✓	0.95790	0.94741	1.63619

The ✓ in the table indicates that the method contains the algorithm.

Table 5. The recall of each sample on binary classification.

Category	Method							
	BLS	PBLS	DBLS	D _e BLS	BELS	PBELS	DBELS	D _e BELS
Benign	0.99983	0.99934	0.99983	0.16582	0.98655	0.95613	0.99554	0.99551
Attack	0.26141	0.18154	0.47439	0.99625	0.86312	0.74888	0.97604	0.95480

Table 6. The recall of each sample on multi-classification.

Category	Method							
	BLS	PBLS	DBLS	D _e BLS	BELS	PBELS	DBELS	D _e BELS
Benign	0.99094	0.99252	0.99787	0.99665	0.99057	0.95042	0.98886	0.95903
Botnet	0	0	0	0.13660	0	0	0	0.94845
ARES	0	0	0.98480	0.98655	1	0	0.99766	0.98830
Brute Force	0.91515	0.79286	0.97416	0.95507	0.87920	0.89234	0.98244	0.94544
DoS/DDoS	0.93683	0	0.99101	0.98787	0.85313	0.97871	0.99093	0.98709
Port Scan	0	0	0	0	0	0	0.86085	0.85613
Web Attack								

4.5.1. Performance Analysis on Binary Classification

In the binary classification task, the traditional BLS structure is less effective in recognizing the CICIDS2017 dataset, mainly due to the high-dimensional redundant information in the dataset that affects the classification performance, resulting in lower accuracy and recall, especially the lack of attention to the minority-class samples in the imbalanced dataset. After the introduction of EQL v2, the accuracy and recall of the BELS model significantly improved, indicating that EQL v2 enhances the ability to recognize minority-class samples in dealing with the imbalance problem. The deep belief network broad learning system (DBLS) model reduces the data redundancy through the DBN dimensionality reduction, which improves the accuracy and recall, which were improved by 0.034 and 0.106, respectively. The D_eBLS model overly focuses on minority-class samples during D_eBN dimensionality reduction, resulting in a decrease in the detection performance for majority-class samples.

DBELS has the highest detection performance in the binary classification task, and the introduction of EQL v2 on top of the DBLS improves the accuracy and recall by 0.077 and 0.249, respectively, indicating that the combination of DBN dimensionality reduction and EQL v2 significantly improves the model performance. Although the D_eBELS model introduces EQL v2 to the D_eBN dimensionality reduction dataset and improves the model performance by adjusting the parameters of positive and negative gradient factors, the accuracy and recall slightly decreased by 0.004 and 0.01, respectively, compared with DBELS, which reflects the negative impact of overlearning abnormal samples on classification performance. For the principal component analysis broad learning system (PBLS) model, after the PCA dimensionality reduction, accuracy and recall decreased by 0.014 and 0.040, respectively, compared with the BLS, mainly due to more information loss during PCA dimensionality reduction and the inability to deal with imbalanced datasets efficiently. The principal component analysis broad equalization learning system (PBELS) model had an accuracy of 0.923 and a recall of 0.853, compared with PBLS, with an improvement of 0.055 and 0.262, indicating that EQL v2 can still significantly improve the detection performance of minority-class samples even when PCA dimensionality reduction is ineffective. However, the PBELS reduced accuracy and recall by 0.070 and 0.106, respectively, compared with DBELS, further demonstrating that DBN dimensionality reduction outperforms PCA dimensionality reduction in binary classification tasks.

4.5.2. Performance Analysis on Multi-Classification

In the multi-classification task, the traditional BLS performed poorly, with an accuracy of 0.975 and a recall of only 0.474, mainly due to the severe imbalance of the dataset that affects the classification performance. The BELS model after the introduction of EQL v2 had an improved recall of 0.051, mainly because EQL v2 improves the data imbalance problem, which allows the model to increase the weight of the attack samples for minority classes and thus improves the detection rate. The DBLS model, after dimensionality reduction by DBN, had an improved accuracy of 0.018 and an improved recall of 0.184, which is attributed to the fact that the DBN dimensionality reduction reduces the data redundancy, mitigates the negative impact of raw data, and improves the detection performance of the model. The D_eBLS model improved recall by 0.02 compared with DBLS, which is attributed to the fact that D_eBN learns more about the minority-class samples, eliminates redundancy between different samples, and at the same time retains the key features of the minority-class samples, which improves the classification and detection performance. The DBELS model improved by 0.018 in DBN dimensionality reduction, and with the introduction of EQL v2, accuracy improved by 0.017 and recall improved by 0.278; compared with DBLS, recall improved from 0.658 to 0.803, which is attributed to the fact that EQL v2 balances the learning weight of the minority-class samples and increases the focus on the minority-class samples.

The accuracy for the D_eBELS model of 0.957 and recall of 0.947 was the best performance in the multi-classification task, and recall was significantly improved by 0.27 compared with D_eBLS, which is because the improved BLS model is better able to deal with unbalanced datasets and enhances the ability to recognize minority-class attacks. The recall of D_eBELS improved by 0.144 compared with DBELS, and the overall classification performance improved significantly. The accuracy of the PBLS model decreased by 0.037 and the recall decreased by 0.176 after PCA dimensionality reduction, since PCA dimensionality reduction leads to more loss of information, which affects the multi-classification effect. The accuracy of the PBELS model was 0.938 and recall was 0.470; compared with PBLS, accuracy and recall were improved by 0.001 and 0.173, respectively, indicating that the introduction of EQL v2 improves the detection rate of minority-class samples. However, compared with D_eBELS, the accuracy and recall of PBELS were reduced by 0.019 and 0.477, respectively, which indicates that DBN dimensionality reduction is much better than the PCA dimensionality reduction model in terms of detection performance, and once again proves the superiority of the proposed model in multi-classification tasks.

4.5.3. Time-Cost Analysis

The comparison of the training times for the different models is shown in Figure 9. For binary classification and multi-classification, the training time of the traditional BLS was about 1.8 s, reflecting the advantage of shallow networks with lower time costs in model classification. The training time of BELS was about 5 s, and the time increase was because the introduction of EQL v2 requires the additional performance of multiple matrix operations. The training time of the DBLS model was about 0.5 s, which is a significant reduction compared with the BLS model, indicating that the high-dimensional data have a significant negative effect on the model training time, and the reduced dimensionality of the dataset reduces the training time to 25% of the original one. The training time of D_eBLS was also about 0.5 s, which indicates that the introduction of EQL v2 after the dimensionality reduction of DBN has almost no effect on the training time. The training time for DBELS was about 1.5 s, which is greatly reduced compared with BELS, thanks to the dimensionality reduction of the DBN model for high-dimensional data. However, there was an increase compared with DBLS due to some additional computations required by the improved BELS model. The training time of D_eBELS was about 1.6 s, which is almost the same as that of DBELS, indicating that the introduction of the low-dimensional dataset with EQL v2 does not have a significant impact on the training time, and there is a slight increase in the cost of time due to a small number of additional computations of the positive and negative gradient matrices in comparison with D_eBLS.

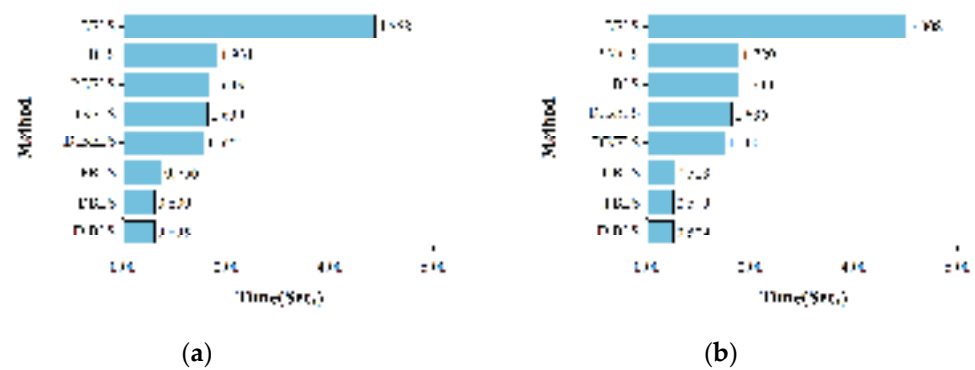


Figure 9. The comparison of training time for different methods: (a) binary classification; (b) multi-classification.

The results show that the hybrid DBN and BLS-based models DBELS and D_eBELS maintained low time consumption compared with the traditional BLS models, following the advantages of the shallow BLS networks that are computationally fast and easy to train. Although the PCA dimensionality reduction model also reduces some of the training time, its detection performance is greatly degraded, again confirming the advantages of the DBN-based dimensionality reduction model. The proposed model is comparable with the traditional deep learning model, which has a significant advantage in time cost consumption, which is conducive to saving a certain amount of computational resources in practical applications.

4.5.4. Results Analysis for Recall of Each Sample on Binary Classification

The detection rate of Benign and Attack samples under the binary classification task is shown in Table 5. The traditional BLS performed poorly in detecting Attack samples with a recall of only 0.261, which is due to the overlearning of the majority-class samples when dealing with imbalanced datasets, resulting in a low detection rate for minority-class samples. With the introduction of the BELS model with EQL v2, the detection rate of Attack significantly improved to 0.863, which significantly improves the identification of minority-class samples. The DBLS model with DBN dimensionality reduction improved the detection rate of Attack by 0.213, which suggests that DBN dimensionality reduction reduces the data redundancy and has a positive impact on the subsequent model classification. In the D_eBLS model, the detection rate of Attack was as high as 0.996, but the detection rate of Benign was only 0.165, which shows that D_eBN dimensionality reduction focuses excessively on minority classes of samples, resulting in a decrease in the recognition performance of the majority of classes of samples.

The DBELS model, after dimensionality reduction by the introduction of EQL v2, had a detection rate of 0.996 for Benign and 0.976 for Attack, which improved the detection rate of the two classes of samples by 0.009 and 0.113, respectively, compared with the BELS model. Because DBN dimensionality reduction reduces the redundant features and improves the classification performance of the model, compared with DBLS, DBELS significantly improved the detection rate of Attack from 0.474 to 0.955, which is attributed to the fact that EQL v2 balances the sample weights so that the model focuses on each class of samples in a more balanced way. The detection rates of Benign and Attack for the D_eBELS model were 0.995 and 0.955, respectively, and compared with D_eBLS, the Benign detection rate was substantially higher and the Attack detection rate was slightly lower, but EQL v2 improved the detection performance by making the model more balanced in focusing on each class of samples. Compared with DBELS, the detection rate of Benign for D_eBELS was almost unchanged, and the detection rate of Attack slightly decreased, which is attributed to the overlearning of minority-class samples brought by D_eBN, but the classification still performed well through the adjustment of positive and negative gradient factors.

The detection rate of Attack for the PBLs model was only 0.182 after the PCA dimensionality reduction, which is a decrease from that of the BLS model of 0.080. PCA dimensionality reduction leads to information loss, which negatively affects the classification detection effect of minority-class samples. The Attack detection rate of the PBELS model was 0.749, which improved by 0.567 compared with the PBLs. The introduction of EQL v2 makes the model focus on each class of sample in a more balanced way, which improves the detection performance of the minority-class samples. Despite the improvement of the PBELS model, its Attack detection rate was still 0.227 lower than that of DBELS, which further proves the superiority of the DBN dimensionality reduction model in detection performance.

4.5.5. Result Analysis for Recall of Each Sample on Multi-Classification

The detection rates of different methods for each type of sample under the multi-classification task are shown in Table 6. The traditional BLS did not work well when dealing with the unbalanced CICIDS2017 dataset, and although it had better detection performance for a larger number of samples (e.g., Benign, DoS/DDoS, Port Scan), it was poor in detecting samples of a few classes (e.g., Botnet ARES, Brute Force, Web Attack). This is because the BLS model is unable to balance the contributions of various classes of samples when dealing with unbalanced datasets, resulting in a very low detection rate for minority-class samples. The introduction of the BELS model with EQL v2 significantly improved the detection rate of Brute Force attacks, which in turn improved Attack identification. However, Botnet ARES and Web Attack were still undetectable because the features of these minority-class samples are highly similar to other samples, which the model is unable to differentiate, leading to false positives. The DBLS model eliminates certain data redundancies after dimensionality reduction by DBN, which enhances the detection performance. The detection rate of Brute Force was improved to 0.985, but Botnet ARES and Web Attack were still not detected. This indicates that dimensionality reduction alone cannot solve the problem of unbalanced datasets, especially for classes with a very small number of samples.

The D_eBLS model further improved the detection performance, and most of the attack types were detected, including Botnet ARES. This is because D_eBN dimensionality reduction learns more about the minority-class samples and preserves their key features, but Web Attack was still undetectable, which requires further focus on the minority-class samples. The DBELS model showed a significant improvement in the detection rate of most attack types compared with BELS and DBLS, especially Web Attack, from 0 to 0.856. This is because DBN dimensionality reduction reduces the data redundancy, while EQL v2 makes the model focus better on the minority-class samples. However, Botnet ARES was still not detected, which suggests that DBN dimensionality reduction is not enough to solve the problem. The D_eBELS model maintained a high detection rate on all attack types. Compared with D_eBELS, the detection rate of Botnet ARES improved from 0.137 to 0.948, and Web Attack improved from 0 to 0.856. This is because the introduction of EQL v2 improves the ability to better recognize and learn the minority-class samples, which significantly improves the detection rate. D_eBELS further improved the detection rate of Botnet ARES compared with DBELS because D_eBN dimensionality reduction preserves the key features of the minority-class samples at a finer granularity and mitigates the data imbalance problem.

The PBLs model using PCA dimensionality reduction only detected the highest number of DoS/DDoS attacks, while Port Scan, Botnet ARES, Brute Force, Web Attack, and other attacks were not detected. Compared with the BLS model, Port Scan could not be detected due to the loss of information caused by PCA dimensionality reduction, which affects the multi-classification effect. After the introduction of EQL v2 to the PBELS model, the detection rate of Port Scan improved to 0.978, but the other few classes of attacks (e.g., Botnet ARES, Brute Force, and Web Attack) still went undetected. This suggests that PCA dimensionality reduction leads to the loss of key features. Compared with

D_eBELS, the detection performance of PBELS was worse, which indicates that DBN-based dimensionality reduction outperforms traditional PCA dimensionality reduction in terms of detection performance.

4.6. Analysis of ROC-AUC

Figure 10 shows the receiver operating characteristic (ROC) curves for the DBELS binary classification. The overall average ROC curve has an area under the curve (AUC) of 0.9949, indicating that the DBELS classifier performed excellently on the entire dataset and is close to being a perfect classifier. Secondly, the ROC curves for the two categories (Attack and Benign) can be observed, each of them with an AUC of 0.9916, which indicates that the classifier also had high performance when considering each category individually, especially when dealing with a minority number of samples. The AUC close to 1 implies that the classifier has a very high true positive rate (TPR) in its predictions for the category while maintaining a relatively low FPR. This means identifying as many true positive samples as possible while keeping the misdiagnosis rate as low as possible. Taken together, the DBELS classifier had excellent performance on the entire imbalance dataset for both minority-class samples and majority-class samples.

The D_eBELS multi-classification ROC curve is shown in Figure 11. The overall average ROC curve has an AUC of 0.9760, which indicates that the classifier performed well on the whole dataset. The Benign, Botnet ARES, Brute Force, DoS/DDoS, and Port Scan detections performed well; their AUC areas were all more than 0.9700, although Web Attack had the lowest AUC area of 0.9100, but it is already better than other models at the same level. Overall, after the introduction of the EQL v2 positive and negative gradient factors, by adjusting the parameter weights, the model can pay more attention to minority classes of samples in the face of imbalanced datasets, thus improving detection performance.

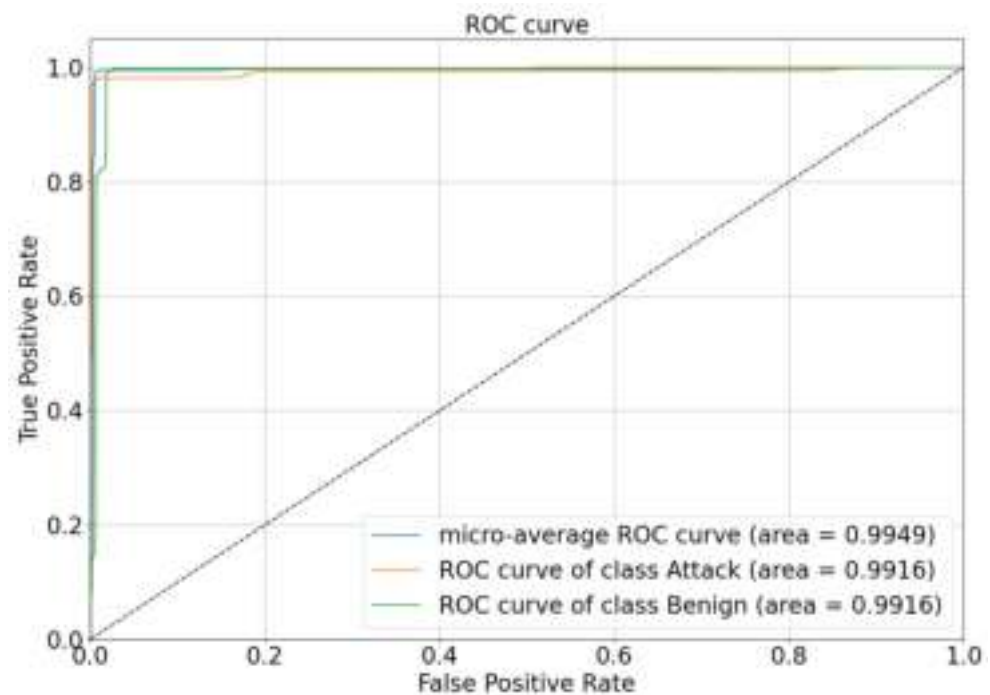


Figure 10. The ROC of DBELS on binary classification.

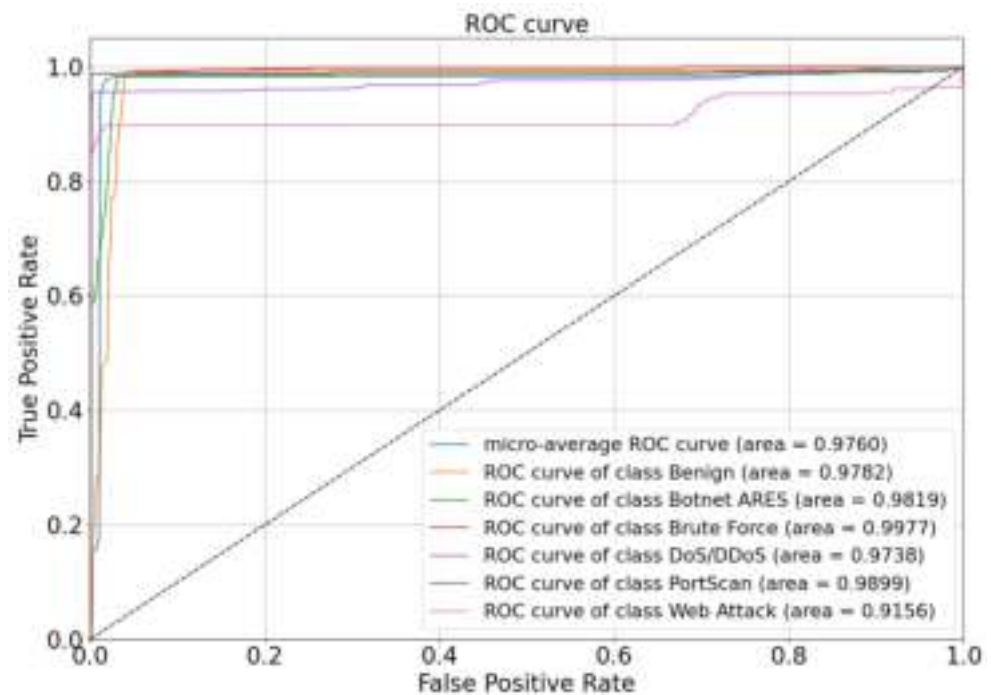


Figure 11. The ROC of D_eBELS on multi-classification.

4.7. Comparison with State-of-the-Art Methods

4.7.1. Binary Classification

The false alarm rate and time cost of all the algorithms were compared on the binary classification task, as shown in Figure 12. For FPR, the proposed model had a lower false alarm rate of 0.014, which was better than the other models. This indicates that the proposed model incorrectly labels normal situations as abnormal less often and correctly identifies and classifies normal situations more often, as compared with other methods. For the model training time, the proposed model required less training time and detection time in the binary classification task, with a time consumption of around 1 s. The time cost of the model is much lower than the deep learning model and has an advantage over other machine learning algorithms, which can significantly reduce the time cost and ensure faster training of the model for large-scale datasets.

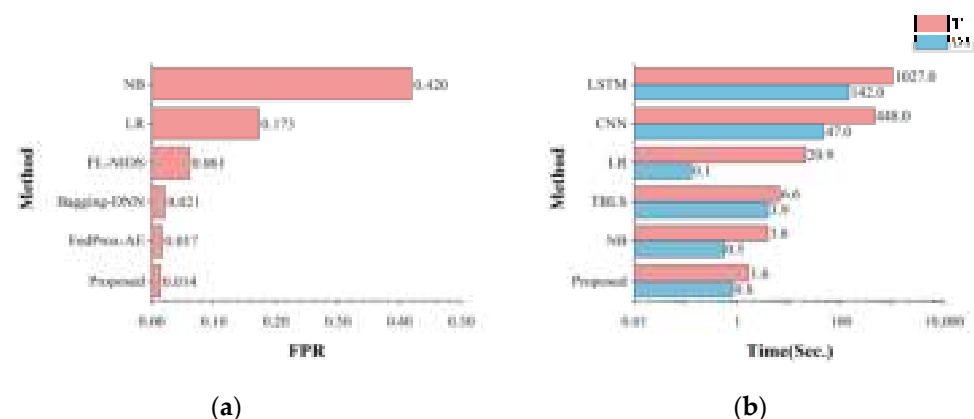


Figure 12. The comparison with state-of-the-art methods on binary classification: (a) the comparison of FPR; (b) the comparison of time cost.

The binary classification detection performance of the different models of recent years is given in Table 7. The proposed model had higher accuracy, recall, and lower FPR in binary classification, which indicates that the proposed model can detect normal and abnormal

behaviors well. Compared with traditional machine learning models, the proposed model outperforms the logistic regression (LR) and naive bayes (NB) in accuracy, recall, and FPR. Compared with other BLS-based models, accuracy and recall are better than TBLS. Compared with deep learning models, the proposed model outperforms most of the models, and the performances of the fusion of statistical deep neural network (FS-DNNs) and multi-objective evolutionary convolutional neural network (MECNN) are slightly higher than the proposed model; however, since both models are based on deep learning models, their model size and training time cost are much higher than the BLS-based model. Considering the bagging ensemble learning deep neural network (Bagging-DNN) model, its accuracy is lower than the proposed model, and its recall is slightly higher than the proposed model, but in addition to the higher space and time complexity brought by its deep learning model, the false alarm rate is also higher than the proposed model. In summary, the proposed model has better detection performance and a lower false alarm rate in binary classification and has obvious advantages over the other models.

Table 7. The comparison with state-of-the-art methods of binary classification.

Method	Accuracy	Recall	FPR
Our study	0.992	0.986	0.014
LR	0.934	0.827	0.173
NB	0.307	0.580	0.420
DCAE	0.925	0.925	-
TBLS(W) [18]	0.982	0.975	-
DNN [18]	0.868	-	-
CNN [18]	0.844	-	-
LSTM [18]	0.365	-	-
2D-CNN [26]	0.980	-	-
CNN-LSTM [27]	0.930	0.768	-
FS-DNN [28]	0.998	0.999	0.012
MECNN [29]	0.998	0.998	-
Bagging-DNN [30]	0.987	0.999	0.021
FL-NIDS [31]	0.943	0.947	0.061
FedProx-AE [32]	0.935	-	0.017

4.7.2. Multi-Classification

The false positive rate and time cost of all the other algorithms on the multi-classification task are shown in Figure 13. The FPR graphs show that the proposed model had only a 0.012 false positive rate, which is lower than the other models. This means that the model can recognize different categories of attacks better and misclassify less often during the multi-classification task and perform the task of detecting and identifying attack samples better. The model required less training time and detection time compared with other models, offering advantages in handling large-scale data and enhancing its suitability for real-world dataset environments.

The multi-classification detection performance of different models in recent years is given in Table 8. The proposed model has higher accuracy and recall and lower FPR under multi-classification task, which indicates that the proposed model can detect both normal and abnormal behaviors well and detect each specific class of attacks better. Comparing the traditional machine learning algorithms, we see that the proposed model outperforms LR and NB in terms of accuracy, recall, and FPR. Compared with other BLS models, the TBLS(W) model has a slightly higher accuracy and recall than the proposed model, since, for the TBLS, the authors only selected the dataset of a single day, Wednesday, for model training and testing. It only accounts for 30% of the total dataset, and the attack types are all DoS/DDoS type attacks, which are better balanced compared with the total dataset, and the detection performance of the model will be affected to some extent.

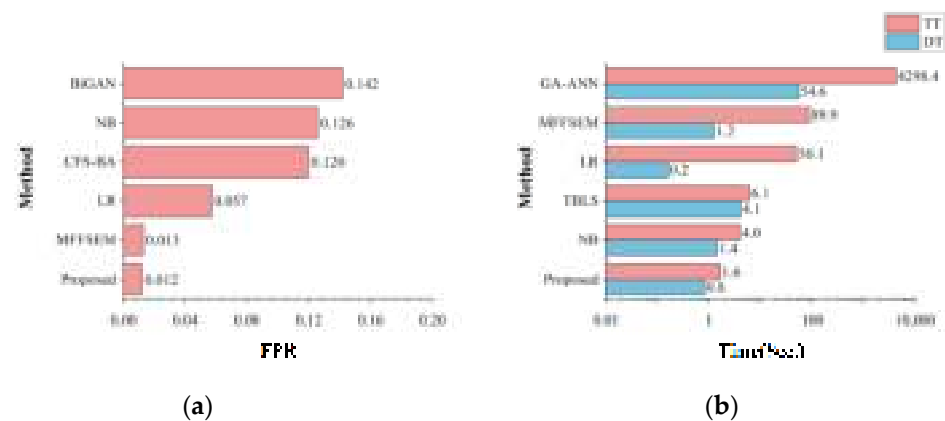


Figure 13. The comparison with state-of-the-art methods on multi-classification: (a) the comparison of FPR; (b) the comparison of time cost.

Table 8. The comparison with state-of-the-art methods on multi-classification.

Method	Accuracy	Recall	FPR
Our study	0.958	0.947	0.012
LR	0.934	0.301	0.057
NB	0.286	0.746	0.126
GA-ANN	0.815	-	-
GSPSO-ANN	0.840	-	-
TBLS(W) [18]	0.978	0.977	-
BiGAN [33]	0.823	0.763	0.142
MFFSEM(W) [2]	0.999	0.999	0.013
DNN [34]	0.946	0.846	-
MECNN [29]	0.997	0.791	-
AdaBoost [35]	0.889	0.234	-
SDAE-SVM [36]	0.954	0.444	-
GRU [37]	0.985	0.742	-
LSTM [38]	0.989	0.748	-
CFS-BA(W) [39]	0.999	0.999	0.120

Compared with the other machine learning algorithms, for the multi-dimensional feature fusion and stacking ensemble mechanism (MFFSEM) and correlation-based feature selection bat algorithm (CFS-BA) models, the detection performances are better than that of the proposed model, but their FPRs are higher than the proposed model, which will lead to them overly misjudging normal behavior as an attack. The two models are also trained with the same dataset of only one day, Wednesday. Compared with the long short-term memory (LSTM) model, its accuracy is slightly higher than the proposed model, but its recall is much lower than the proposed model, which means that the LSTM model fails to recognize the attacks well. In summary, compared with other models, the proposed model has advantages such as better detection performance and a lower false alarm rate when performing the multi-classification task.

5. Model Validation

Network traffic in real environments is large and complex; to evaluate the DBELS model better and simulate the effect of real environments, we selected the CICIDS2018 dataset to test and validate the model's performance in different data environments. The CICIDS2018 dataset is a widely used dataset in cybersecurity research and was created by the Canadian institute for cybersecurity research, which simulates real-world network traffic, including normal traffic and many types of attacks. Compared to the CICIDS2017 dataset, its size has increased by about four times and contains more than eight million traffic records; however, it also has a serious data imbalance. The dataset contains over

eight million traffic records with up to 80 features per sample. These data cover both normal traffic and a wide range of attack traffic, with sample types mainly including Benign, DoS/DDoS, Brute Force, Botnet and Infiltration. The diversity and complexity of the sample types in the CICIDS 2018 dataset make the dataset a better reflection of real-world cyber threats, which makes it possible to comprehensively evaluate the performance and generalization capabilities of intrusion detection systems.

The model validation experiments using the CICIDS2018 dataset included data preprocessing and data dimensionality reduction, and the form of the operations remained consistent with CICIDS2017. The models for their comparison experiments include BLS, DT, NB, MLP, and CNN. To ensure the validity of the experiments, the data preprocessing is the same, and all the experiments are conducted in a consent environment. The evaluation metrics for the comparison experiment included accuracy, recall, false positive rate, and training time. The performance of different models under the binary classification task is shown in Table 9, and the performance under the multiple classification task is shown in Table 10.

Table 9. The performance of different methods on binary classification for the CICIDS2018 dataset.

Method	Evaluation Metric			
	Accuracy	Recall	FPR	Time(s)
Our study	0.981	0.956	0.044	5.295
BLS	0.958	0.931	0.069	18.826
DT	0.917	0.862	0.138	16.770
NB	0.791	0.742	0.258	2.556
MLP	0.963	0.933	0.067	861.788
CNN	0.957	0.926	0.074	949.476

Table 10. The performance of different methods on multi-classification for the CICIDS2018 dataset.

Method	Evaluation Metric			
	Accuracy	Recall	FPR	Time(s)
Our study	0.946	0.923	0.016	4.731
BLS	0.953	0.702	0.031	12.729
DT	0.891	0.499	0.060	17.266
NB	0.828	0.860	0.047	2.694
MLP	0.934	0.485	0.046	1025.025
CNN	0.966	0.785	0.026	916.550

The results show that our proposed model outperforms most other models in accuracy, recall, FPR and time cost under different classification tasks. For the binary classification task, compared to the traditional BLS structure, accuracy and recall are improved by 0.023 and 0.025, respectively, and the false alarm rate is decreased by 0.025. Compared to the traditional machine learning algorithms DT and NB, both of them have a significant improvement in the detection performance, while the false alarm rate is decreased significantly. For the deep learning algorithms MLP and CNN, the detection performance is improved and the model training time is reduced significantly. For the multi-classification task, the proposed model also maintains high detection performance and low model training time, and compared to the NB model, it has a significant advantage in detection performance, although the model training time is slightly improved. Compared with the CNN model, although the accuracy slightly decreases by 0.02, it is better than the CNN model in terms of recall, FPR, and model training time.

From the above experimental results, it can be concluded that the proposed model can still show better detection performance when facing the CICIDS2018 dataset with a larger and more complex scale, which further indicates that the model has a certain degree of scalability and adaptability. The scale of this dataset is greatly increased compared to the CICIDS2017 dataset; however, the proposed model can still maintain low time consumption, which further verifies that the model has a certain degree of adaptability and lightness for real complex environments. Our proposed model consistently demonstrates strong detection performance and low time consumption across various datasets, whether for binary or multi-class classification. This efficiency is crucial for real-world applications, where maintaining high accuracy and speed is essential. Additionally, the model's adaptability to different data environments and robustness against diverse threats enhances its practical value. Its scalability ensures it can handle increasing data volumes, making it a versatile solution in dynamic cybersecurity landscapes.

6. Conclusions

In this paper, based on the BLS model and introducing the EQL v2 reweighting idea, we propose a DBELS model, which aims to solve the problem of the traditional BLS facing low detection rates for high-dimensional and imbalanced datasets in the field of NIDSs, as well as the low recognition of minority-class attacks. First, the model fully learns the large-scale high-dimensional dataset through DBN, fully trains each RBM structure, captures the optimal structure and key information of the high-dimensional dataset, and represents it as an optimal low-dimensional dataset. Subsequently, the BELS model is applied to learn the classification of the low-dimensional dataset, and the learning degree of the model on the minority-class samples is improved by adjusting the positive and negative gradient factors to increase the classification weight of the model and enhance the recognition of the minority-class attacks. The DBELS is fully tested experimentally using the CICIDS2017 dataset and our proposed model outperforms other algorithms in accuracy, recall, FPR, and time. Finally, the utility of the proposed model is further tested and validated with the CICIDS2018 dataset. The results show that our proposed DBELS model has significant advantages for solving the problem of high-dimensional and imbalanced network intrusion detection data and alleviates the problems of long training time of intrusion detection systems and low recognition of minority-class attacks, which makes it a practical intrusion detection method. We plan to introduce initialization strategies and flow-learning techniques in the future to improve the stability and detection performance of the BLS. Furthermore, we plan to use advanced algorithms to enhance the robustness of the model and improve its adaptability and anti-interference capabilities.

Author Contributions: Conceptualization, M.D.; methodology, M.D. and C.S.; software, M.D., C.S. and Y.K.; validation, M.D.; investigation, C.S., M.D. and S.F.; resources, X.Z.; data curation, C.S. and S.F.; writing—original draft preparation, M.D. and H.X.; writing—review and editing, M.D. and H.X.; visualization, M.D. and Y.K.; supervision, X.Z. and H.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (62276091).

Data Availability Statement: Data will be made available on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Anderson, J.P. *Computer Security Threat Monitoring and Surveillance*; Technical Report; James P. Anderson Company: Washington, DC, USA, 1980.
2. Zhang, H.; Li, J.-L.; Liu, X.-M.; Dong, C. Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection. *Future Gener. Comput. Syst.* **2021**, *122*, 130–143. [[CrossRef](#)]
3. Sain, H.; Purnami, S.W. Combine sampling support vector machine for imbalanced data classification. *Procedia Comput. Sci.* **2015**, *72*, 59–66. [[CrossRef](#)]

4. Thapa, N.; Liu, Z.; Kc, D.B.; Gokaraju, B.; Roy, K. Comparison of machine learning and deep learning models for network intrusion detection systems. *Future Internet* **2020**, *12*, 167. [\[CrossRef\]](#)
5. Namakshenas, D.; Yazdinejad, A.; Dehghantanha, A.; Srivastava, G. Federated quantum-based privacy-preserving threat detection model for consumer internet of things. *IEEE Trans. Consum. Electron.* **2024**. [\[CrossRef\]](#)
6. Almiani, M.; AbuGhazleh, A.; Al-Rahayfeh, A.; Atiewi, S.; Razaque, A. Deep recurrent neural network for IoT intrusion detection system. *Simul. Model. Pract. Theory* **2020**, *101*, 102031. [\[CrossRef\]](#)
7. Wu, P. *Deep Learning for Network Intrusion Detection: Attack Recognition with Computational Intelligence*; UNSW: Sydney, Australia, 2020.
8. Yazdinejad, A.; Dehghantanha, A.; Karimipour, H.; Srivastava, G.; Parizi, R.M. A Robust Privacy-Preserving Federated Learning Model Against Model Poisoning Attacks. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 6693–6708. [\[CrossRef\]](#)
9. Yazdinejad, A.; Dehghantanha, A.; Srivastava, G.; Karimipour, H.; Parizi, R.M. Hybrid privacy preserving federated learning against irregular users in next-generation Internet of Things. *J. Syst. Archit.* **2024**, *148*, 103088. [\[CrossRef\]](#)
10. Chen, C.P.; Liu, Z. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 10–24. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Gong, X.; Zhang, T.; Chen, C.P.; Liu, Z. Research review for broad learning system: Algorithms, theory, and applications. *IEEE Trans. Cybern.* **2021**, *52*, 8922–8950. [\[CrossRef\]](#)
12. Li, Z.; Rios, A.L.G.; Xu, G.; Trajković, L. Machine Learning Techniques for Classifying Network Anomalies and Intrusions. In Proceedings of the 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 26–29 May 2019; pp. 1–5.
13. Wu, T.; Fan, H.; Zhu, H.; You, C.; Zhou, H.; Huang, X. Intrusion detection system combined enhanced random forest with SMOTE algorithm. *EURASIP J. Adv. Signal Process.* **2022**, *2022*, 39. [\[CrossRef\]](#)
14. Zhang, B.; Liu, Z.; Jia, Y.; Ren, J.; Zhao, X. Network intrusion detection method based on PCA and Bayes algorithm. *Secur. Commun. Netw.* **2018**, *2018*, 1914980. [\[CrossRef\]](#)
15. Li, Z.; Batta, P.; Trajkovic, L. Comparison of Machine Learning Algorithms for Detection of Network Intrusions. In Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 7–10 October 2018; pp. 4248–4253.
16. Rios, A.L.G.; Li, Z.; Xu, G.; Alonso, A.D.; Trajković, L. Detecting Network Anomalies and Intrusions in Communication Networks. In Proceedings of the 2019 IEEE 23rd International Conference on Intelligent Engineering Systems (INES), Gödöllő, Hungary, 25–27 April 2019; pp. 000029–000034.
17. Rios, A.L.G.; Li, Z.; Bekshentayeva, K.; Trajković, L. Detection of Denial of Service Attacks in Communication Networks. In Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 12–14 October 2020; pp. 1–5.
18. Li, J.; Zhang, H.; Liu, Z.; Liu, Y. Network intrusion detection via tri-broad learning system based on spatial-temporal granularity. *J. Supercomput.* **2023**, *79*, 9180–9205. [\[CrossRef\]](#)
19. Ahmad, T.; Aziz, M.N. Data preprocessing and feature selection for machine learning intrusion detection systems. *ICIC Express Lett.* **2019**, *13*, 93–101.
20. Hao, X.; Jiang, Z.; Xiao, Q.; Wang, Q.; Yao, Y.; Liu, B.; Liu, J. Producing More with Less: A GAN-Based Network Attack Detection Approach for Imbalanced Data. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 5–7 May 2021; pp. 384–390.
21. Tan, J.; Lu, X.; Zhang, G.; Yin, C.; Li, Q. Equalization Loss v2: A New Gradient Balance Approach for Long-Tailed Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 1685–1694.
22. Shen, Z.; Zhang, Y.; Chen, W. A bayesian classification intrusion detection method based on the fusion of PCA and LDA. *Secur. Commun. Netw.* **2019**, *2019*, 6346708. [\[CrossRef\]](#)
23. Salo, F.; Nassif, A.B.; Essex, A. Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection. *Comput. Netw.* **2019**, *148*, 164–175. [\[CrossRef\]](#)
24. Hinton, G.E.; Osindero, S.; Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Belarbi, O.; Khan, A.; Carnelli, P.; Spyridopoulos, T. An Intrusion Detection System Based on Deep Belief Networks. In Proceedings of the International Conference on Science of Cyber Security, Shimane, Japan, 10–12 August 2022; pp. 377–392.
26. Andresini, G.; Appice, A.; Malerba, D. Nearest cluster-based intrusion detection through convolutional neural networks. *Knowl. Based Syst.* **2021**, *216*, 106798. [\[CrossRef\]](#)
27. Kim, A.; Park, M.; Lee, D.H. AI-IDS: Application of deep learning to real-time Web intrusion detection. *IEEE Access* **2020**, *8*, 70245–70261. [\[CrossRef\]](#)
28. Thakkar, A.; Lohiya, R. Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System. *Inf. Fusion* **2023**, *90*, 353–363. [\[CrossRef\]](#)
29. Chen, Y.; Lin, Q.; Wei, W.; Ji, J.; Wong, K.-C.; Coello, C.A.C. Intrusion detection using multi-objective evolutionary convolutional neural network for Internet of Things in Fog computing. *Knowl. Based Syst.* **2022**, *244*, 108505. [\[CrossRef\]](#)
30. Thakkar, A.; Lohiya, R. Attack classification of imbalanced intrusion data for IoT network using ensemble learning-based deep neural network. *IEEE Internet Things J.* **2023**, *10*, 11888–11895. [\[CrossRef\]](#)

31. Mulyanto, M.; Faisal, M.; Prakosa, S.W.; Leu, J.-S. Effectiveness of focal loss for minority classification in network intrusion detection systems. *Symmetry* **2020**, *13*, 4. [\[CrossRef\]](#)
32. Idrissi, M.J.; Alami, H.; El Mahdaoui, A.; El Mekki, A.; Oualil, S.; Yartaoui, Z.; Berrada, I. Fed-anids: Federated learning for anomaly-based network intrusion detection systems. *Expert Syst. Appl.* **2023**, *234*, 121000. [\[CrossRef\]](#)
33. Yao, W.; Shi, H.; Zhao, H. Scalable anomaly-based intrusion detection for secure Internet of Things using generative adversarial networks in fog environment. *J. Netw. Comput. Appl.* **2023**, *214*, 103622. [\[CrossRef\]](#)
34. Jose, J.; Jose, D.V. Deep learning algorithms for intrusion detection systems in internet of things using CIC-IDS 2017 dataset. *Int. J. Electr. Comput. Eng. (IJECE)* **2023**, *13*, 1134–1141. [\[CrossRef\]](#)
35. Rojas, R. AdaBoost and the super bowl of classifiers a tutorial introduction to adaptive boosting. *Freie Univ. Berl. Tech. Rep.* **2009**, *1*, 1–6.
36. Lv, Z.; Qiao, L.; Li, J.; Song, H. Deep-learning-enabled security issues in the internet of things. *IEEE Internet Things J.* **2020**, *8*, 9531–9538. [\[CrossRef\]](#)
37. Assis, M.V.; Carvalho, L.F.; Lloret, J.; Proença Jr, M.L. A GRU deep learning system against attacks in software defined networks. *J. Netw. Comput. Appl.* **2021**, *177*, 102942. [\[CrossRef\]](#)
38. Imrana, Y.; Xiang, Y.; Ali, L.; Abdul-Rauf, Z. A bidirectional LSTM deep learning approach for intrusion detection. *Expert Syst. Appl.* **2021**, *185*, 115524. [\[CrossRef\]](#)
39. Zhou, Y.; Cheng, G.; Jiang, S.; Dai, M. Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Comput. Netw.* **2020**, *174*, 107247. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.