electronics

MDPI

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

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.

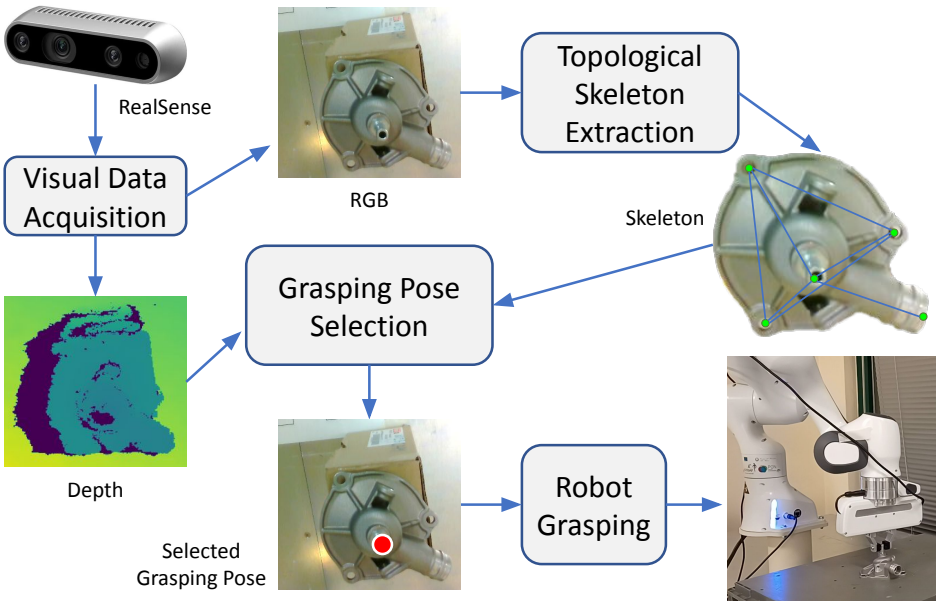


Figure 1. Functional architecture of the proposed approach.

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, $\mathcal{F}_c$.

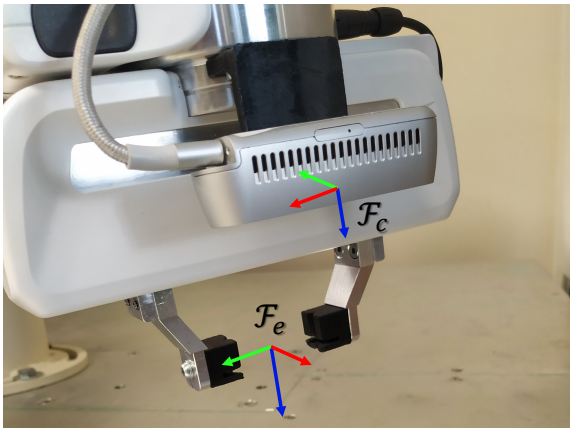


Figure 2. End-effector and camera reference frames.

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

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

where $ReLU6(x) = min(max(x, 0), 6)$ 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.

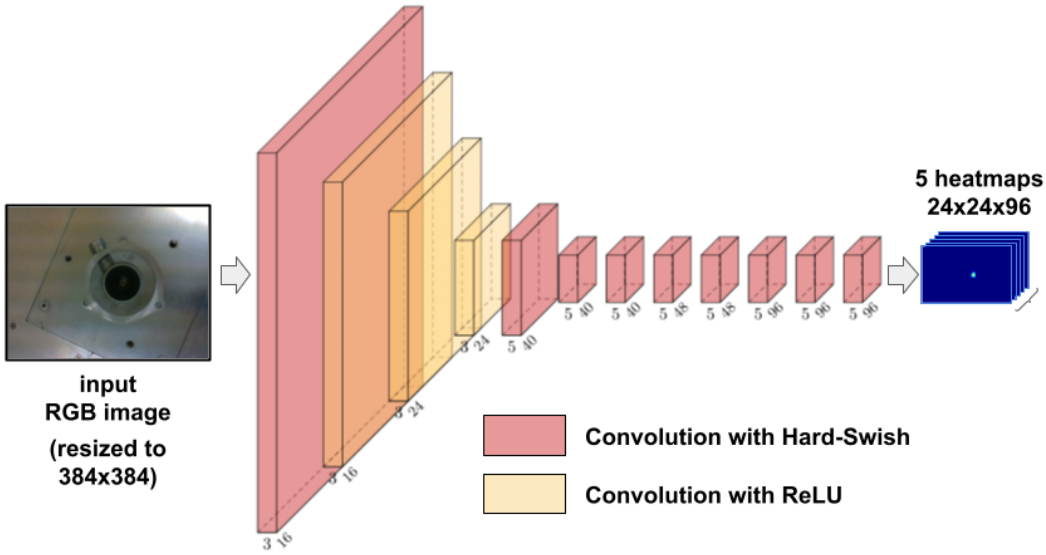


Figure 3. MobileNetV3 architecture.

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 o 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.

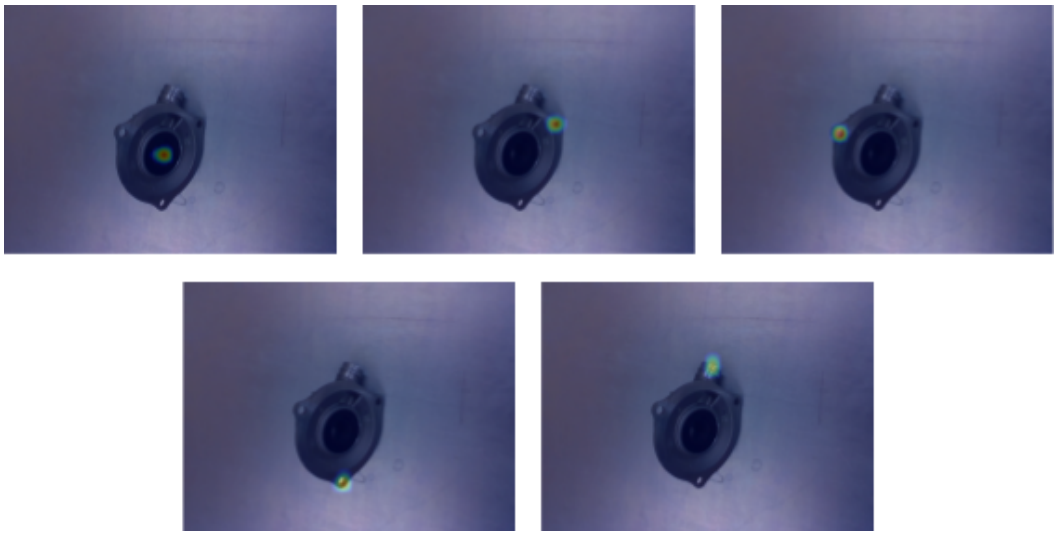


Figure 4. Heatmap examples for cast iron crankcase oil separator cover. There are five heatmaps corresponding to the considered keypoints.

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.

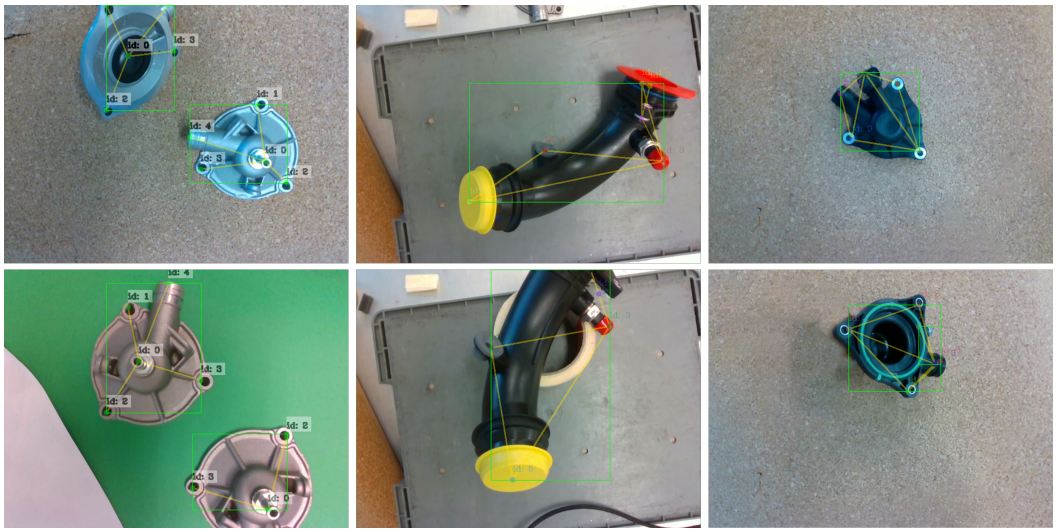


Figure 5. TS extraction examples on different objects: the cast iron crankcase oil separator cover on the left, the air pipe in the middle, and the plastic crankcase oil separator cover on the right. Our approach is robust to different views of the same object, to photochromic changes, and partial occlusions.

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], $\boldsymbol{T}^f_{g_j}$, $j = 1, \ldots, N_g$, 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

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

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 $\boldsymbol{T}^f_{t_i}$. For each grasping reference frame, the relative pose with respect to the ith plane can be determined as

$$\boldsymbol{T}^{t_i}_{g_j} = \left(\boldsymbol{T}^f_{t_i}\right)^{-1} \boldsymbol{T}^f_{g_j}\ . \tag{2}$$

Thus, for each grasping point, a list of N_t transformation matrices, $\boldsymbol{T}^{t_i}_{g_j}$, 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, $\boldsymbol{T}^f_{g_j}$ $(j = 1, \ldots, N_g)$
Output: $\boldsymbol{T}^{t_i}_{g_j}$ $(i = 1, \ldots, N_t; j = 1, \ldots, N_g)$

```
for each triple t_i of keypoints in TS_N do
    Compute T^f_{t_i}
end for
for i = 1, ..., N_t do
    for j = 1, ..., N_g do
        T^{t_i}_{g_j} = (T^f_{t_i})^{-1} T^f_{g_j}
    end for
end for
return T^{t_i}_{g_j}
```

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, $\boldsymbol{p}^c_{cc}$, 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, $\boldsymbol{T}^c_{t_k}$, and to select the homogeneous transformation matrix, $\boldsymbol{T}^{t_k}_{g_{cc}}$, 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

$$\boldsymbol{T}^c_{g_{cc}} = \boldsymbol{T}^c_{t_k} \boldsymbol{T}^{t_k}_{g_{cc}}\ .$$

This procedure is summarized in Algorithm 2.

Let us define the homogeneous transformation matrix $\boldsymbol{T}^e_c$, 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 : ${}^{obj}\boldsymbol{T}_{g_j}^{t_i}$ for each object
Output: $\boldsymbol{T}_{g_{cc}}^{c}$

Get image from camera
Detect current object from image
$TS_C \leftarrow$ extract TS
$\mathcal{T} = \boldsymbol{T}_{g_j}^{t_i}\Big|_{i=1,\ldots,N_t;\ j=1,\ldots,N_g} \leftarrow$ get the list of transformations relative to the current object
$\boldsymbol{p}_{cc}^{c} \leftarrow$ extract the grasping point closest to the camera
if *visible_keypoints* ≥ 3 **then**
    $t_k \leftarrow$ extract a triple of visible keypoints in TS_C
    Compute $\boldsymbol{T}_{t_k}^{c}$
    Extract $\boldsymbol{T}_{g_{cc}}^{t_k}$ from $\mathcal{T}$
else
    move the robot to a different point of view
    GO TO 1
end if
$\boldsymbol{T}_{g_{cc}}^{c} \leftarrow \boldsymbol{T}_{t_k}^{c}\,\boldsymbol{T}_{g_{cc}}^{t_k}$
return $\boldsymbol{T}_{g_{cc}}^{c}$

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

$$\boldsymbol{T}_{g_{cc}} = \boldsymbol{T}_{e}\,\boldsymbol{T}_{c}^{e}\,\boldsymbol{T}_{g_{cc}}^{c}\ , \tag{3}$$

where $\boldsymbol{T}_e$ 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, $\hat{\boldsymbol{p}}_g$. More in detail, the end-effector desired position, $\boldsymbol{p}_{e,d}(t)$, is defined as

$$\boldsymbol{p}_{e,d}(t) = \begin{cases} \boldsymbol{p}_0 + \dfrac{s_1(t)}{\|\boldsymbol{p}_a - \boldsymbol{p}_0\|}(\boldsymbol{p}_a - \boldsymbol{p}_0) & \text{for } 0 \leq t \leq t_a \\ \\ \boldsymbol{p}_a + \dfrac{s_2(t)}{\|\hat{\boldsymbol{p}}_g - \boldsymbol{p}_a\|}(\hat{\boldsymbol{p}}_g - \boldsymbol{p}_a) & \text{for } t_a < t \leq t_f \end{cases} . \tag{4}$$

where $\boldsymbol{p}_0$ is the view position and $\boldsymbol{p}_a$ is the approach point position, $s_1(t)$ ($s_2(t)$) is the *arc length* form $\boldsymbol{p}_0$ to $\boldsymbol{p}_a$ (from $\boldsymbol{p}_a$ to $\hat{\boldsymbol{p}}_g$). 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, $\widehat{\boldsymbol{R}}_g$, 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 $\boldsymbol{p}_e(t)$ and $\boldsymbol{R}_e(t)$ the end-effector position and orientation, respectively, and with $\boldsymbol{R}_{e,d}(t)$ the end-effector desired orientation. The robot joint velocity references, $\dot{\boldsymbol{q}}_r(t)$, are computed as

$$\dot{\boldsymbol{q}}_r(t) = \boldsymbol{J}^{\dagger}(\boldsymbol{q}(t))(\dot{\boldsymbol{v}}_{e,d}(t) + \boldsymbol{K}\boldsymbol{e}(t)), \tag{5}$$

where $\boldsymbol{J}^{\dagger}(\boldsymbol{q}(t))$ denotes the right pseudo-inverse of the robot Jacobian matrix, $\boldsymbol{K} \in \mathbb{R}^{6\times 6}$ is a positive definite matrix gain, $\boldsymbol{v}_{e,d} = \begin{bmatrix} \dot{\boldsymbol{p}}_{e,d}^{\mathrm{T}} & \boldsymbol{\omega}_{e,d}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ is the desired end-effector linear and angular velocity, and $\boldsymbol{e}$ is the tracking error defined as

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{p}_{e,d} - \boldsymbol{p}_e \\ \eta_e\boldsymbol{\epsilon}_{e,d} - \eta_{e,d}\boldsymbol{\epsilon}_e - \boldsymbol{S}(\boldsymbol{\epsilon}_{e,d})\boldsymbol{\epsilon}_e \end{bmatrix}, \tag{6}$$

where $\mathcal{Q}_e = \{\eta_e, \boldsymbol{\epsilon}_e\}$ and $\mathcal{Q}_{e,d} = \{\eta_{e,d}, \boldsymbol{\epsilon}_{e,d}\}$ are the unit quaternion extracted from $\boldsymbol{R}_e$ and $\boldsymbol{R}_{e,d}$, respectively, and $\boldsymbol{S}(\cdot)$ is the skew-symmetric matrix operator performing the cross product [22].

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

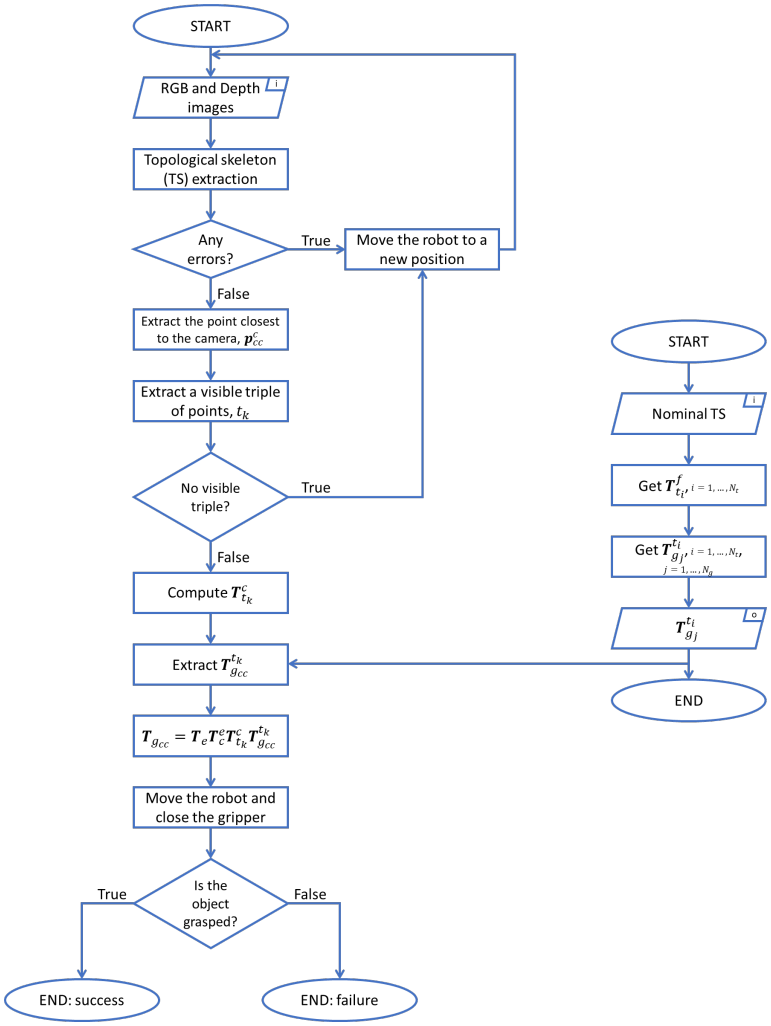


Figure 7. Flowchart representation of the whole process.

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:

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

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

$$\boldsymbol{e}_p^e = \boldsymbol{p}_g^e - \hat{\boldsymbol{p}}_g^e, \tag{8}$$

$$\boldsymbol{e}_\phi^e = \boldsymbol{\phi}_g^e - \hat{\boldsymbol{\phi}}_g^e, \tag{9}$$

where $\boldsymbol{p}_g^e$ is the actual grasping position while $\hat{\boldsymbol{p}}_g^e$ is the estimate provided by the visual algorithm. Regarding the orientation, $\boldsymbol{\phi}_g^e$ ($\hat{\boldsymbol{\phi}}_g^e$) 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.

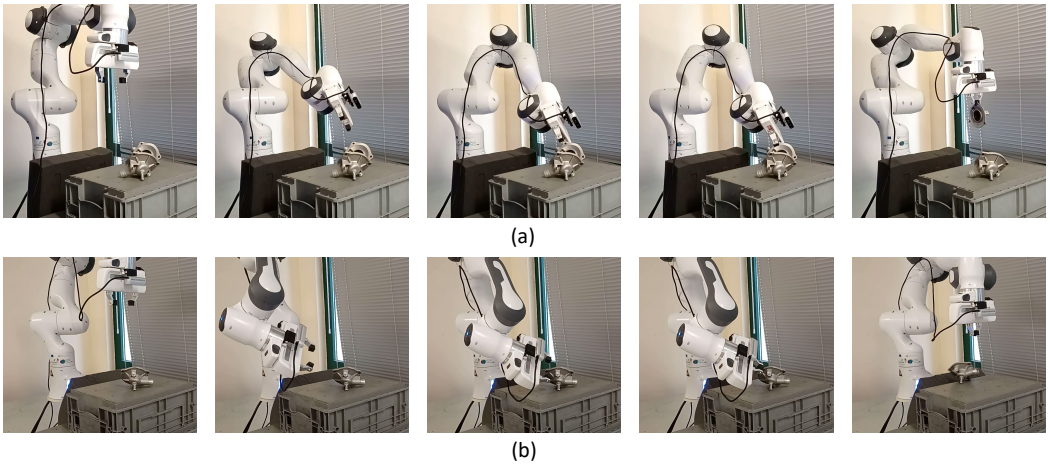


Figure 8. Snapshots of two grasping cases. (**a**) Successful grasp. (**b**) Failure.

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	$e^e_{p_x}$ [mm]	$e^e_{p_y}$ [mm]	$e^e_{\phi_x}$ [deg]	$e^e_{\phi_y}$ [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	$e^e_{p_x}$ [mm]	$e^e_{p_y}$ [mm]	$e^e_{\phi_x}$ [deg]	$e^e_{\phi_y}$ [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	$e^e_{p_x}$ [mm]	$e^e_{p_y}$ [mm]	$e^e_{\phi_x}$ [deg]	$e^e_{\phi_y}$ [deg]	$e^e_{\phi_z}$ [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

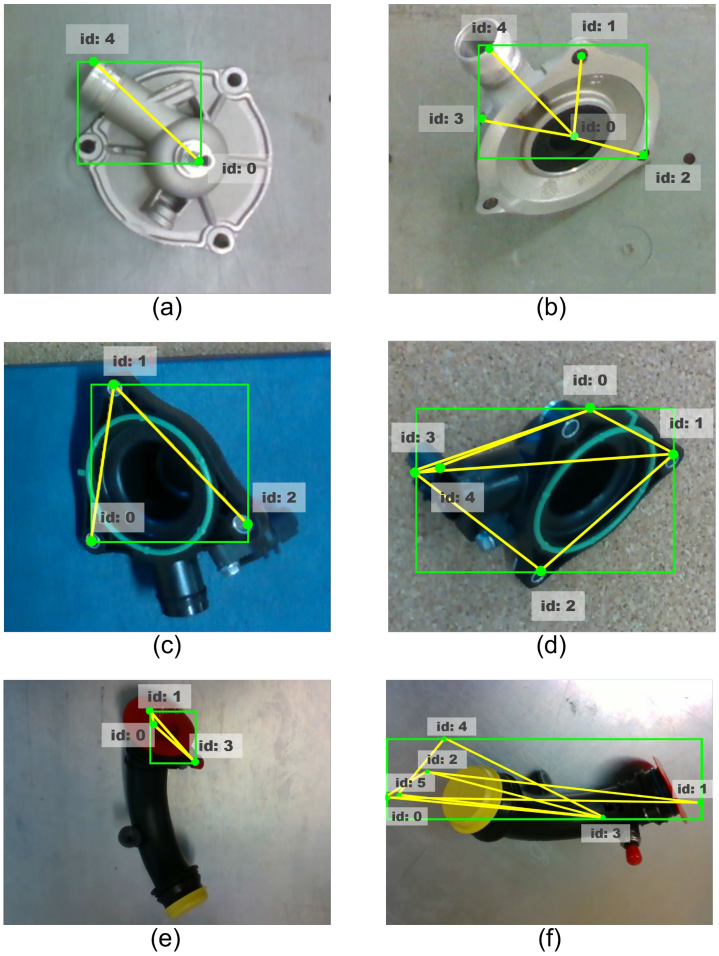


Figure 9. Examples of missing (**a**,**e**) and inaccurate (**b**,**d**,**f**) keypoint detection in TS. Example of missing keypoint detection that can lead to a successful object grasping (**c**).

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.; Collowet, 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.; Sciavicco, 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).

electronics

MDPI

Article

Improved Active Disturbance Rejection Control for Permanent Magnet Synchronous Motor

Zhiwei Huang [1], Yuanhao Cheng [1,2,*], Si Chen [1,2], Xuhui Zhang [3], Jiawei Xiang [1,2] and Sun'an Wang [4]

[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

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.

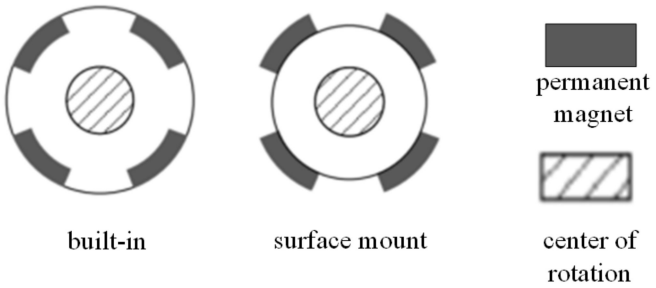


Figure 1. Schematic diagram of PMSM rotor structures.

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:

$$\begin{cases} u_{\mathrm{a}} = \frac{\mathrm{d}\psi_{\mathrm{a}}}{\mathrm{d}t} + R_{\mathrm{s}} i_{\mathrm{a}} \\ u_{\mathrm{b}} = \frac{\mathrm{d}\psi_{\mathrm{b}}}{\mathrm{d}t} + R_{\mathrm{s}} i_{\mathrm{s}} \\ u_{\mathrm{c}} = \frac{\mathrm{d}\psi_{\mathrm{c}}}{\mathrm{d}t} + R_{\mathrm{s}} i_{\mathrm{c}} \end{cases} \tag{1}$$

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

$$\begin{cases} u_{\mathrm{d}} = \frac{\mathrm{d}\psi_{\mathrm{d}}}{\mathrm{d}t} + R_{\mathrm{s}} i_{\mathrm{d}} - \omega_{\mathrm{e}} \psi_{\mathrm{q}} \\ u_{\mathrm{q}} = \frac{\mathrm{d}\psi_{\mathrm{q}}}{\mathrm{d}t} + R_{\mathrm{s}} i_{\mathrm{q}} + \omega_{\mathrm{e}} \psi_{\mathrm{d}} \end{cases} \tag{2}$$

The stator magnetism-chain equation is:

$$\begin{cases} \psi_{\mathrm{d}} = L_{\mathrm{d}} i_{\mathrm{d}} + \psi_{\mathrm{f}} \\ \psi_{\mathrm{q}} = L_{\mathrm{q}} i_{\mathrm{q}} \end{cases} \tag{3}$$

Bringing Equation (3) into (2) gives:

$$\begin{cases} u_{\mathrm{d}} = \frac{\mathrm{d}\psi_{\mathrm{d}}}{\mathrm{d}t} + R_{\mathrm{s}} i_{\mathrm{d}} - \omega_{\mathrm{e}} L_{\mathrm{q}} i_{\mathrm{q}} \\ u_{\mathrm{q}} = \frac{\mathrm{d}\psi_{\mathrm{q}}}{\mathrm{d}t} + R_{\mathrm{s}} i_{\mathrm{q}} + \omega_{\mathrm{e}} (L_{\mathrm{d}} i_{\mathrm{d}} + \psi_{\mathrm{f}}) \end{cases} \tag{4}$$

where $u_{\mathrm{a}}, u_{\mathrm{b}}, u_{\mathrm{c}}$ are the stator winding's three-phase voltages, with V as the unit; R_{s} is the stator resistance, with Ω as the unit; $i_{\mathrm{a}}, i_{\mathrm{b}}, i_{\mathrm{c}}$ are the stator winding's three-phase current, with A as the unit; $\psi_{\mathrm{a}}, \psi_{\mathrm{b}}, \psi_{\mathrm{c}}$ 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; $L_{\mathrm{d}}, L_{\mathrm{q}}$ 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:

$$J \frac{\mathrm{d}\omega_{\mathrm{m}}}{\mathrm{d}t} = T_{\mathrm{e}} - B\omega_{\mathrm{m}} - T \tag{5}$$

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:

$$T_{\mathrm{e}} = \frac{3}{2} p_n \left[\psi_{\mathrm{f}} i_{\mathrm{q}} + (L_{\mathrm{d}} - L_{\mathrm{q}}) i_{\mathrm{d}} i_{\mathrm{q}} \right] \tag{6}$$

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:

$$J \frac{\mathrm{d}\omega_{\mathrm{m}}}{\mathrm{d}t} = \frac{3}{2} p_{\mathrm{n}} \left[\psi_{\mathrm{f}} i_{\mathrm{q}} + (L_{\mathrm{d}} - L_{\mathrm{q}}) i_{\mathrm{d}} i_{\mathrm{q}} \right] - B\omega_{\mathrm{m}} - T \tag{7}$$

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):

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

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

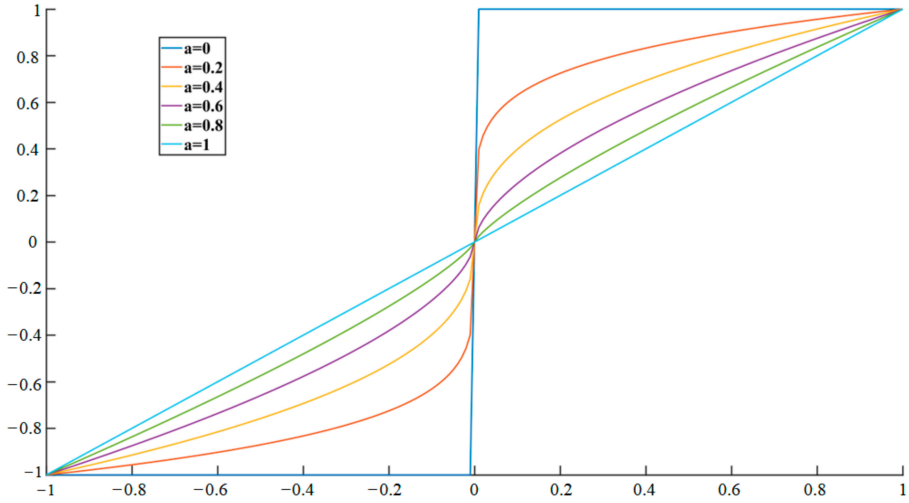


Figure 2. Curves of $fal(e, \alpha, \delta)$ function under different α values.

From the perspective of convergence, the terminal attractor $|e|^{\alpha} sign(e)$ 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, α, δ)* 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:

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

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:

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

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

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

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

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

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

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

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:

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

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

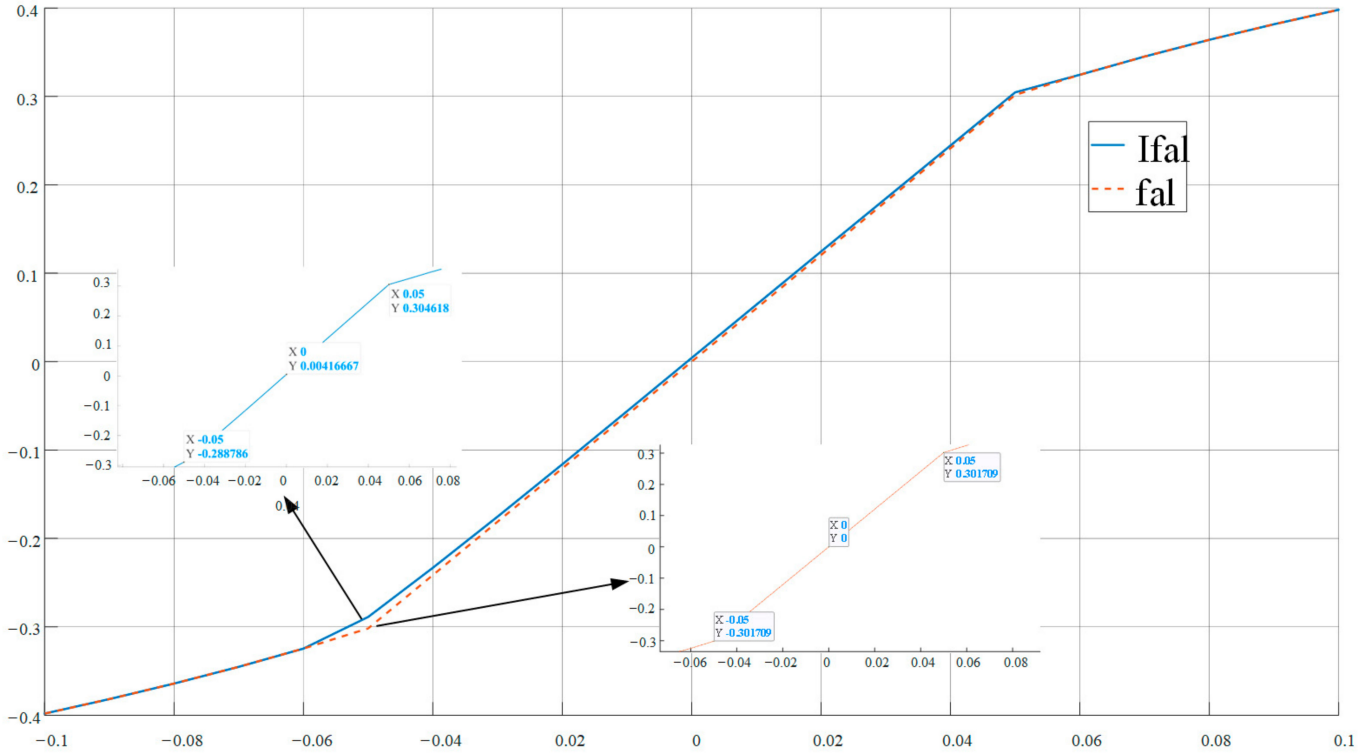


Figure 3. Curves of *fal*(*e*, *α*, *δ*) function before and after improvement.

It can be seen from the graph that the *Ifal*(*e*, *α*, *δ*) 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:

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

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:

$$\begin{cases} e_1 = z_{11} - z_{21} \\ u(t) = -\frac{z_{22}}{b} + k_2 Ifal(e_1, \alpha_1, \delta_1) \end{cases} \tag{16}$$

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.

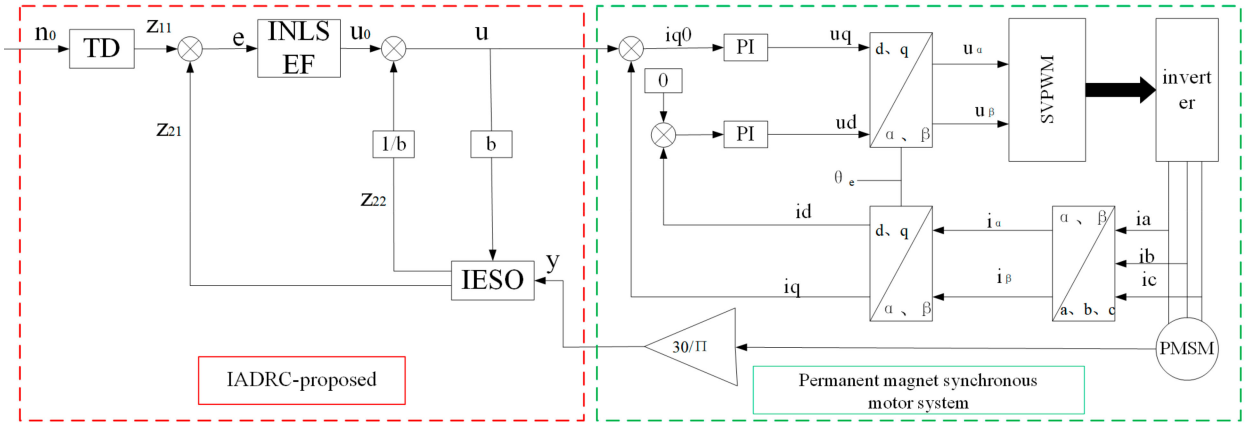


Figure 4. Diagram outlining PMSM's IADRC-proposed speed control system.

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 LD (H)	8.5×10^{-3}
Stator cross-axis inductance LQ (H)	8.5×10^{-3}
Permanent magnet flux linkage ψ_f (Wb)	0.175
Inertia J (kg·m^2/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.

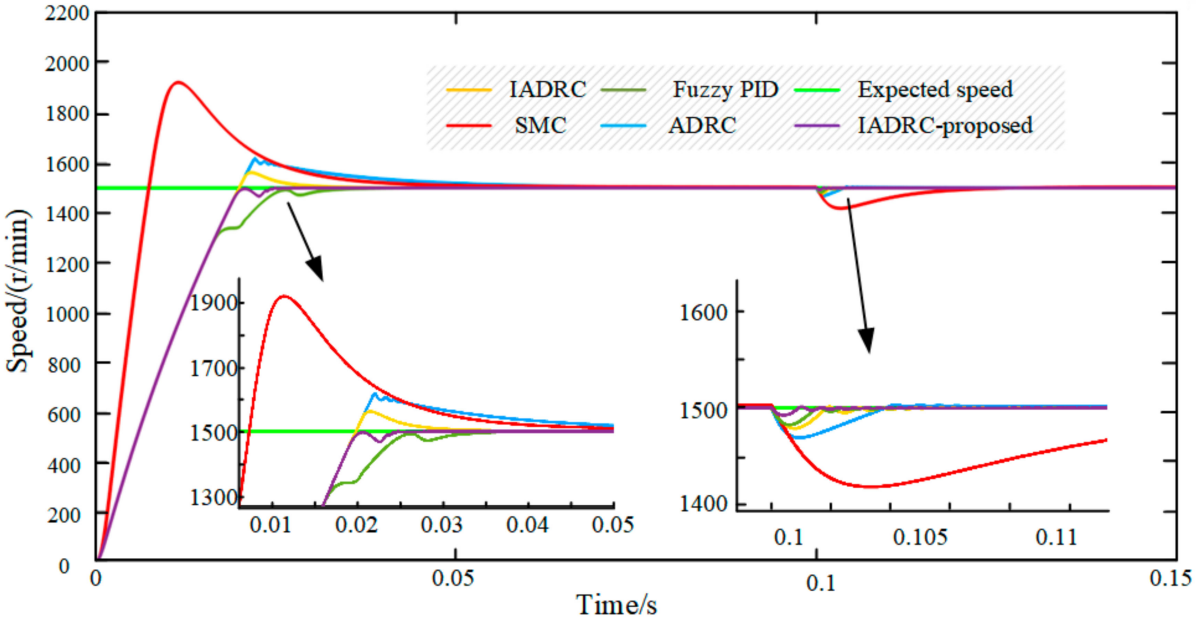


Figure 5. Speed response curves of 5 algorithms without Gaussian noise.

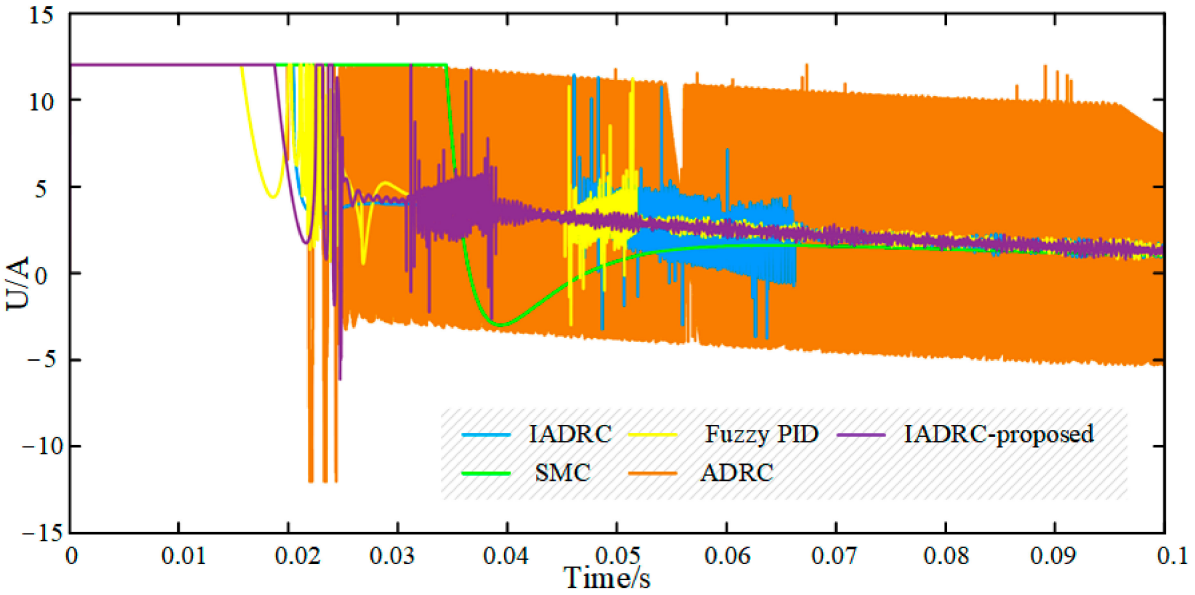


Figure 6. The plant control inputs of 5 algorithms without Gaussian noise.

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.

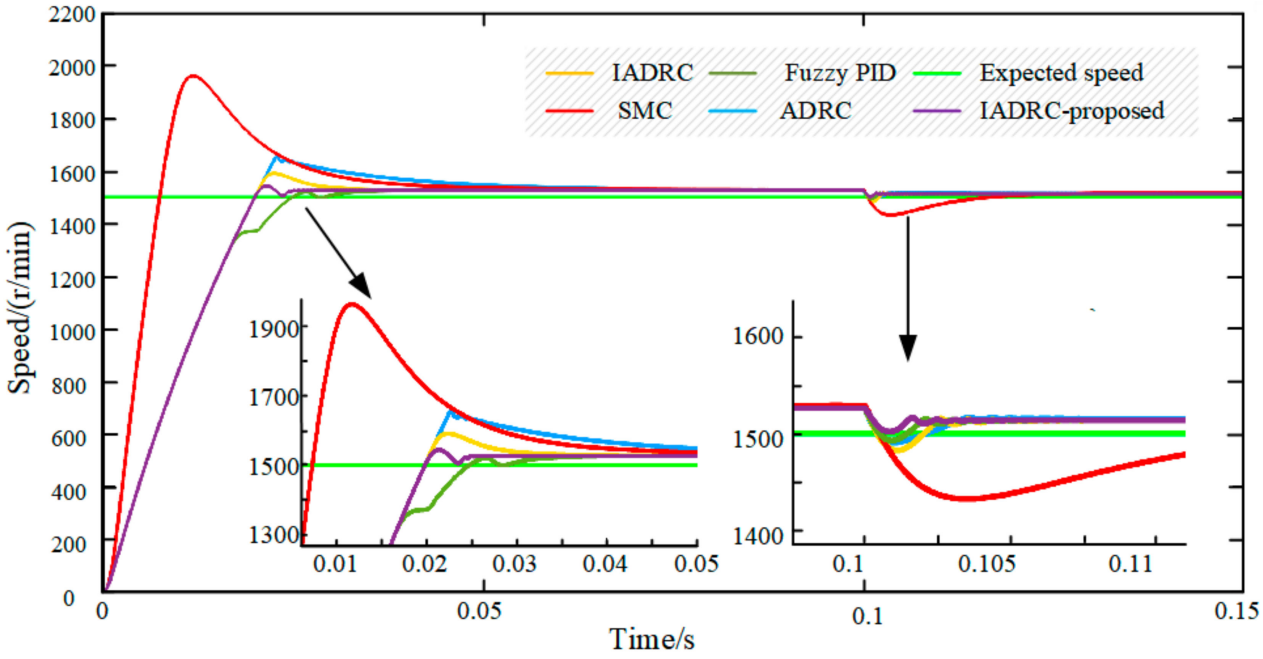


Figure 7. Speed response curves of 5 algorithms under Gaussian noise.

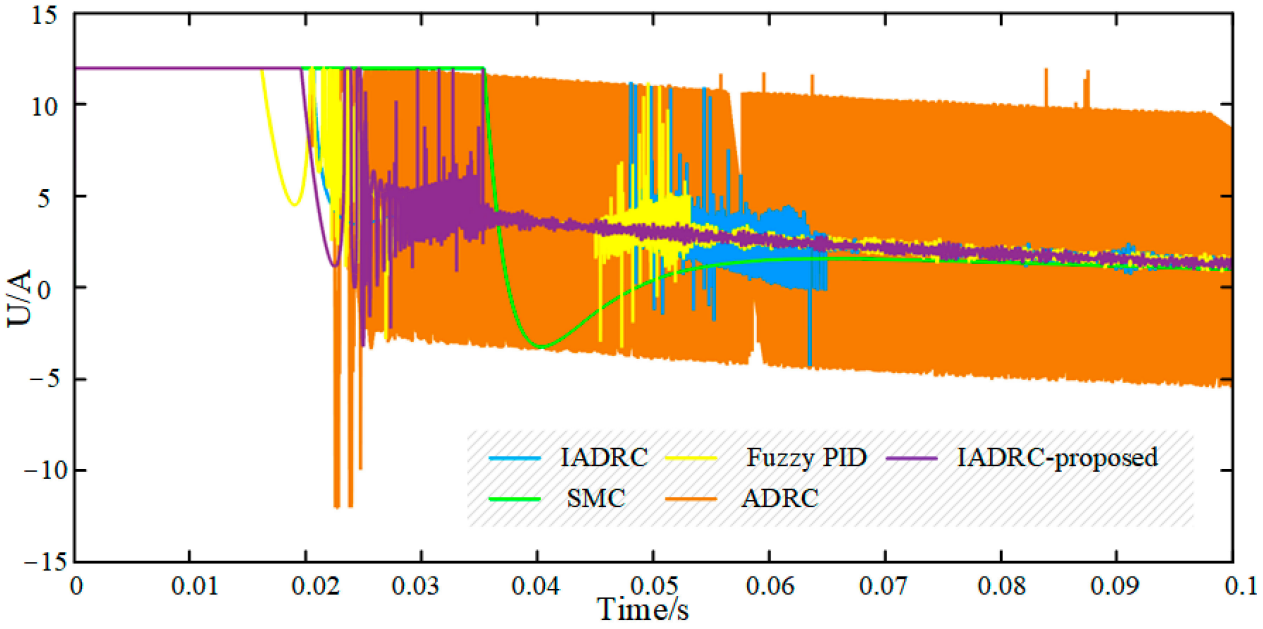


Figure 8. The plant control inputs of 5 algorithms under Gaussian noise.

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.

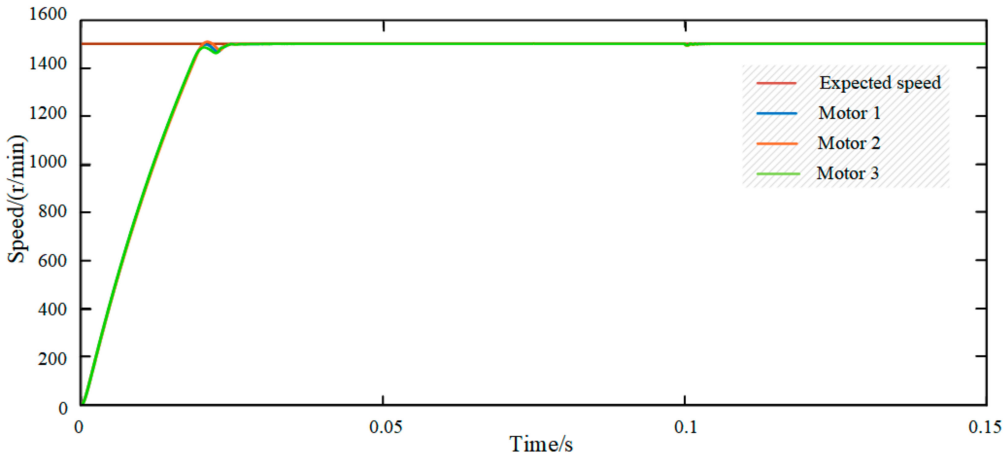


Figure 9. Motor speed curves with different resistances.

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. Electrif.* **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. Electrif.* **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.

electronics

MDPI

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: Niusha Shafiabady and Jianlong Zhou

Received: 11 June 2024
Revised: 23 July 2024
Accepted: 24 July 2024
Published: 31 July 2024

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 $\mathbf{x} \in \boldsymbol{\mathcal{X}} \subseteq \mathbb{R}^d$ and some additional protected attributes $\mathbf{s} \in \boldsymbol{\mathcal{S}} \subseteq \mathbb{R}^p$, 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 $h : \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{S}} \rightarrow \mathcal{Y}$ from a class $\mathbf{H}$. The protected attributes $\mathbf{s} \in \boldsymbol{\mathcal{S}}$ in our study are assumed to be binary with a special value n denoting the unprivileged group. For example, $\boldsymbol{\mathcal{S}}$ could be race and n "non-White"; therefore, the binary nature of $\boldsymbol{\mathcal{S}}$ 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 $\boldsymbol{\mathcal{X}}$, and protected attribute set $\boldsymbol{\mathcal{S}}$ with domain value $\{w, n\}$. The discrimination in $\mathcal{D}$ with respect to the group $\boldsymbol{\mathcal{S}} = n$ 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 $\boldsymbol{\mathcal{S}}$ [21]. FPR and TPR with respect to protected attribute $\mathbf{s} \in \boldsymbol{\mathcal{S}}$ 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

```
baselineCount ← sum of entries in 'Largest_Category' of X_train
totalCount ← number of entries in X_train
baselineProportion ← baselineCount / totalCount
balancedData ← initialize an empty dataset
categories ← list of column names in X_train starting with 'combined_category_'
for each category in categories do
    categoryData ← select entries in X_train where category = 1
    categoryData ← combine categoryData with corresponding labels from y_train
    categoryCount ← number of entries in categoryData
    targetCount ← integer part of totalCount × baselineProportion
    if categoryCount < targetCount then
        sampledData ← sample targetCount from categoryData with replacement
        balancedData ← append sampledData to balancedData
    else
        balancedData ← append categoryData to balancedData
    end if
end for
return balancedData
```

Algorithm 2 Protected-Category Proportional Sampling

```
targetSamples ← 5000
sampledBalanced ← initialize an empty data set
for each column in new_categories.columns do
    categoryRows ← select rows in new_categories where column = 1
    sampledRows ← sample targetSamples entries from categoryRows with replacement
    for each col in oneHotEncodedBalanced.columns do
        sampledRows[col] ← 0
    end for
    sampledRows[column] ← 1
    sampledBalanced ← append sampledRows to sampledBalanced
end for
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)

procedure CUSTOMSMOTE(*data*, *n_samples*)
  syntheticSamples ← zero matrix of size (*n_samples*× number of columns in *data*)
  nn ← NearestNeighbors(*n_neighbors* = 7).fit(*data*)
  neighbors ← *nn*.kneighbors(*data*, return_distance = *False*)
  for *i* ← 1 **to** *n_samples* **do**
    sampleIdx ← random integer from 0 to (number of rows in *data* − 1)
    nnIdx ← random choice from *neighbors*[*sampleIdx*, 1 :]
    diff ← *data*[*nnIdx*] − *data*[*sampleIdx*]
    weight ← random number from uniform distribution between 0 and 1
    syntheticSamples[*i*] ← *data*[*sampleIdx*] + *weight* × *diff*
  end for
  return *syntheticSamples*
end procedure

Algorithm 4 Generate Balanced Synthetic Labels

procedure GENBALSYNTHLABELS(*n_samplesNeeded*, *balanceRatio*)
  *nClass*1 ← int(*n_samplesNeeded* × *balanceRatio*)
  *nClass*0 ← *n_samplesNeeded* − *nClass*1
  syntheticLabels ← [0] × *nClass*0 + [1] × *nClass*1
  SHUFFLE(*syntheticLabels*) ▷ Randomly shuffle the labels
  return *syntheticLabels*
end procedure

Algorithm 5 Protected-category SMOTE

balancedDataList ← initialize an empty list
for each *category* in *categories* **do**
  categorySubset ← select rows in *train_data* s.t. 'combined_category' == *category*
  features ← remove 'class', 'combined_category' from *categorySubset*
  nSamples ← *max_size*− number of rows in *categorySubset*
  if *nSamples* > 0 **then**
    syntheticFeatures ← PCSmote(*features*, *nSamples*)
    syntheticLabels ← GenBalSynthLabels(*nSamples*, *balanceRatio*)
    syntheticFeatures['*class*'] ← *syntheticLabels*
    syntheticFeatures['*combined_category*'] ← *category*
    categorySubsetBalanced ← concatenate *categorySubset* and *syntheticFeatures*
  else
    categorySubsetBalanced ← *categorySubset*
  end if
  append *categorySubsetBalanced* to *balancedDataList*
end for
balancedData ← 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

procedure PCADASYNCATEGORIES(*data*, *labels*, *n_samplesNeeded*, *n_neighbors*)
  $n\_neighbors \leftarrow n\_neighbors + 1$ ▷ Including the data point itself
  $nn \leftarrow \text{NearestNeighbors}(n\_neighbors).\text{fit}(data)$
  $distances, indices \leftarrow nn.\text{kneighbors}(data)$
  $densities \leftarrow$ zero array of length$(data)$
  for $i \leftarrow 0$ **to** length$(data) - 1$ **do**
    $current\_category \leftarrow labels[i]$
    $neighbor\_indices \leftarrow indices[i][1:]$ ▷ Skip the self index
    $densities[i] \leftarrow \text{SUM}(labels[neighbor\_indices] == current\_category)$
  end for
  $weights \leftarrow 1/(densities + 1)$ ▷ Add 1 to prevent division by zero
  $weights \leftarrow weights/\text{SUM}(weights)$ ▷ Normalize weights
  $syntheticSamples \leftarrow$ empty list
  $sampleIndices \leftarrow$ random choice with replacement from length$(data)$ using $weights$
  for each idx **in** $sampleIndices$ **do**
    $baseIdx \leftarrow idx$
    $neighborIdx \leftarrow \text{RANDOMCHOICE}(indices[baseIdx][1:])$
    $diff \leftarrow data[neighborIdx] - data[baseIdx]$
    $syntheticSample \leftarrow data[baseIdx] + \text{RANDOM}() \times diff$
    append $syntheticSample$ to $syntheticSamples$
  end for
  return array$(syntheticSamples)$
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

balancedDataList ← initialize an empty list
for each *category* in *categories* **do**
    categorySubset ← select from *new_data*3 s.t. 'combined_category' == *category*
    features ← remove 'class', 'combined_category' from *categorySubset*
    categoryLabels ← extract 'combined_category' from *categorySubset*
    nSamplesNeeded ← *max_size* minus number of rows in *categorySubset*
    if *nSamplesNeeded* > 0 **then**
        syntheticFeatures ← PCAdasyn(*features*, *categoryLabels*, *nSamplesNeeded*)
        syntheticLabels ← GenBalSynthLabels(*nSamplesNeeded*, *balanceRatio*)
        syntheticFeatures['*class*'] ← *syntheticLabels*
        syntheticFeatures['*combined_category*'] ← *category*
        categorySubsetBalanced ← concatenate *categorySubset* with *syntheticFeatures*
    else
        categorySubsetBalanced ← *categorySubset*
    end if
    append *categorySubsetBalanced* to *balancedDataList*
end for
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.

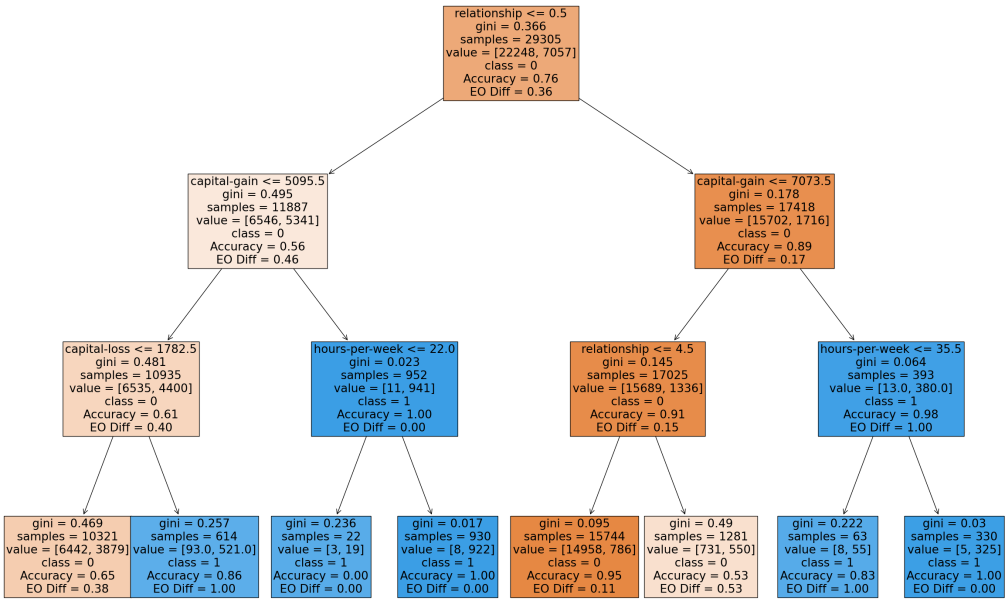


Figure 1. Example decision tree trained on Adult Income with no sampling.

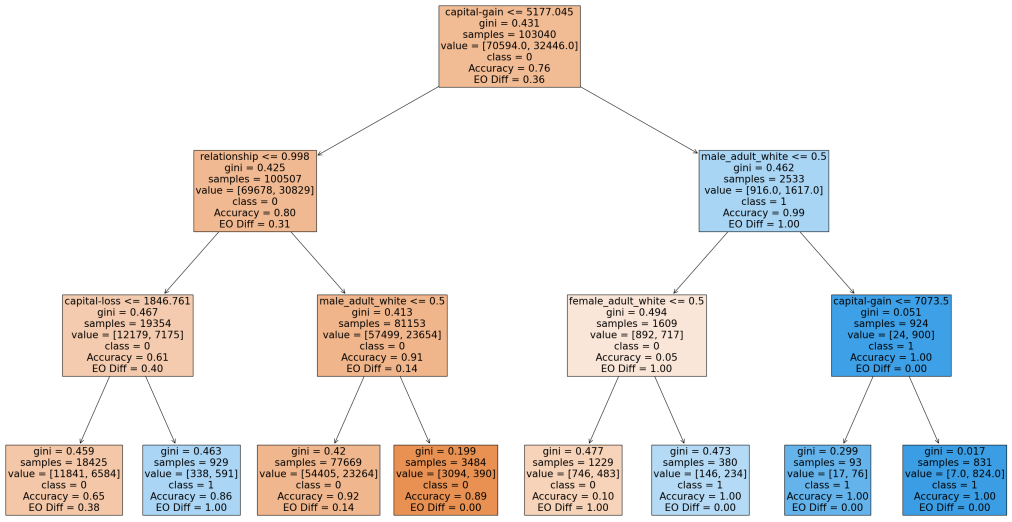


Figure 2. Decision tree of PC-ADASYN on adult income.

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.

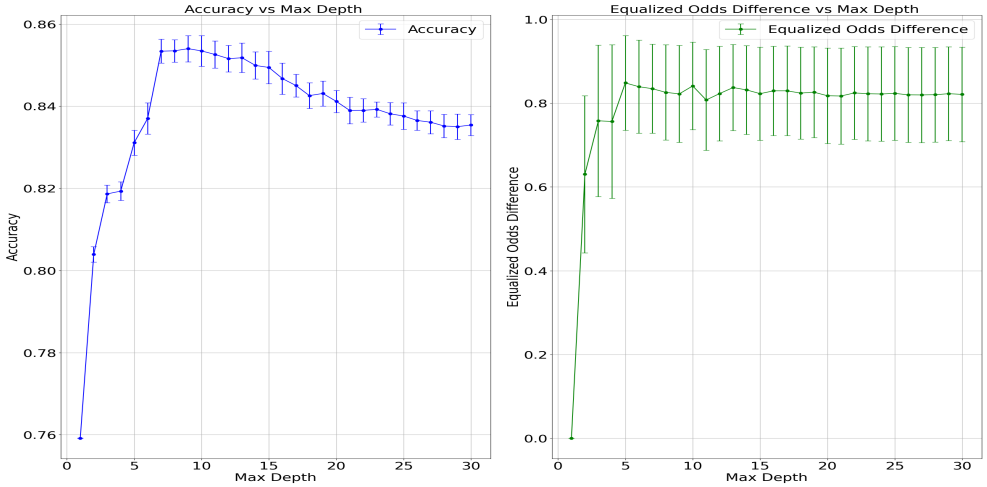


Figure 3. Plots of Adult Income using no sampling, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

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 oversampling 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.

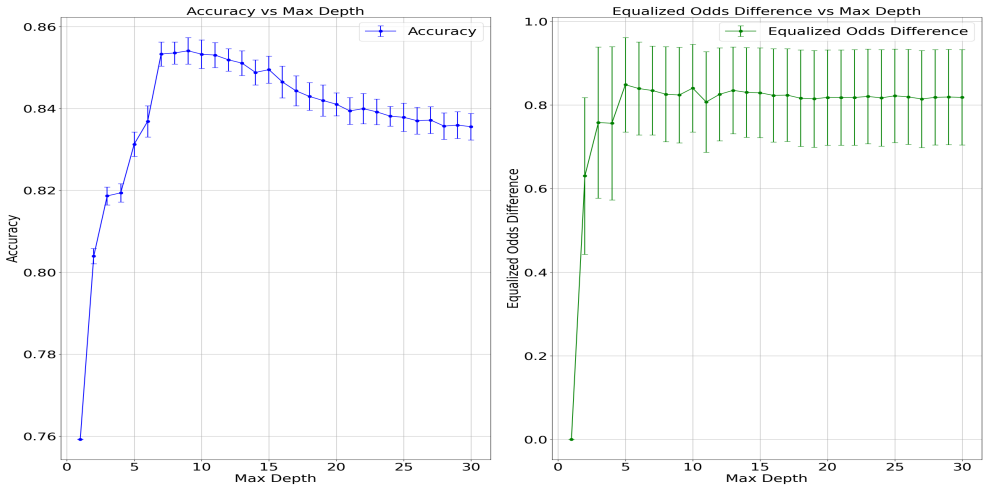


Figure 4. Plots of Adult Income using oversampling, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

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.

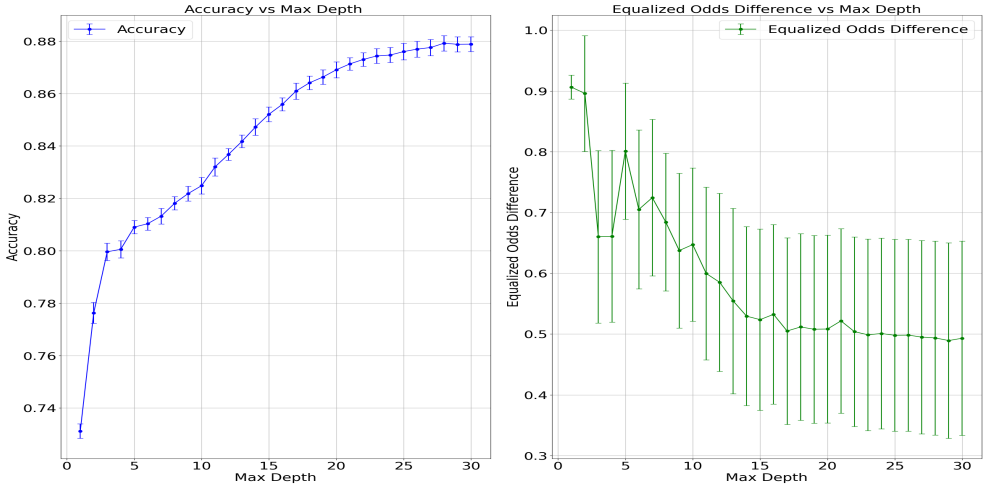


Figure 5. Plots of Adult Income using proportional sampling, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

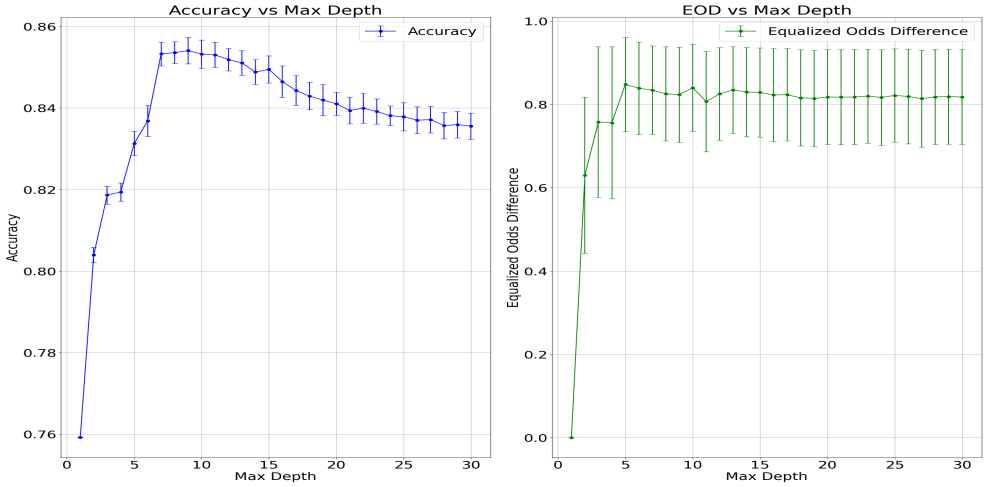


Figure 6. Plots of Adult Income using PC-SMOTE, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

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.

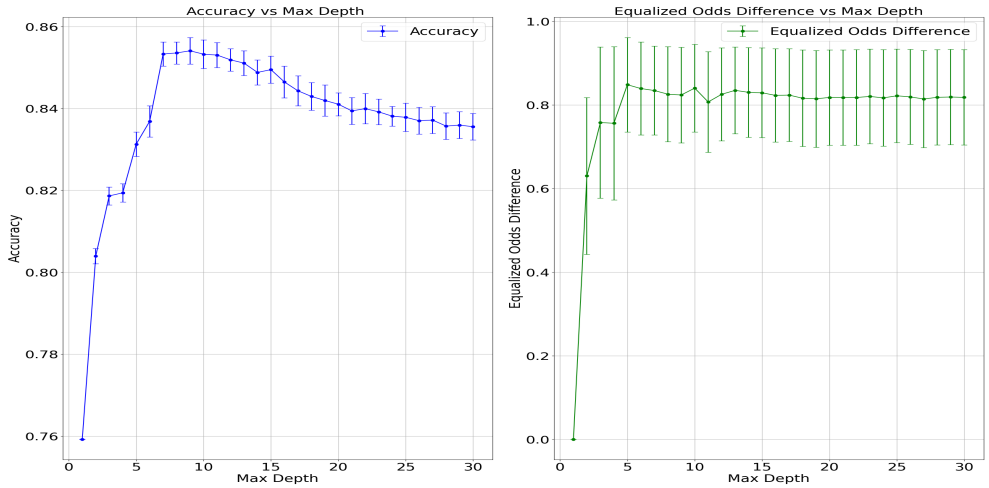


Figure 7. Plots of Adult Income using PC-ADASYN, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

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.; Cemballi, 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.; Žliobaitè, 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 Massachusets, 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, ICDSST 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.

electronics

MDPI

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 [1]

[1] Department of Technology Management for Innovation, School of Engineering, The University of Tokyo, Tokyo 113-8656, Japan
[2] 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

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 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].

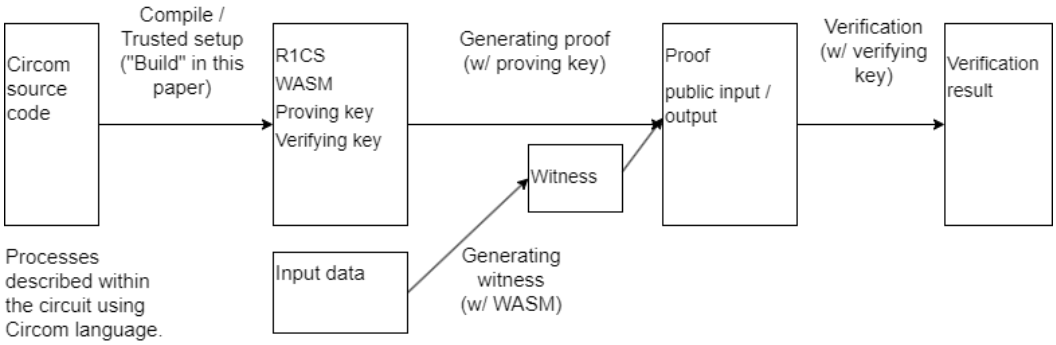


Figure 1. Zero-knowledge proof using circom.

4. Proposed System

Figure 2 presents an overview of the proposed system.

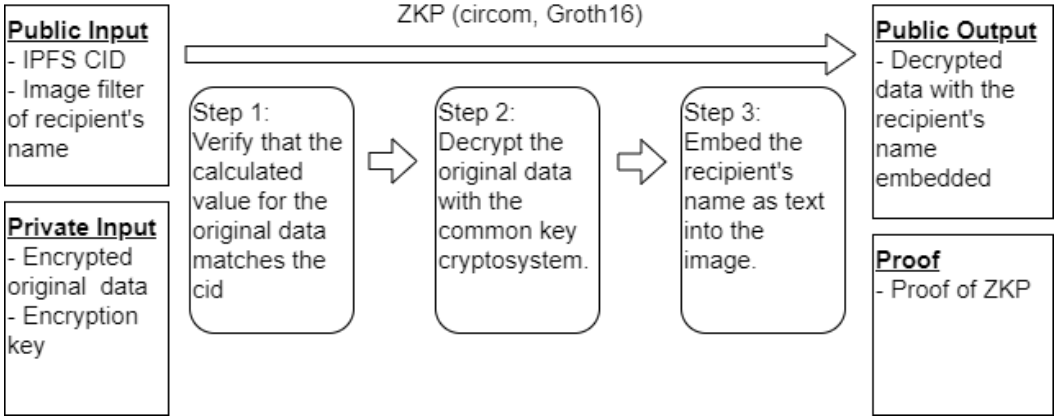


Figure 2. Process flow of ZKP: an example of image data processing.

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.

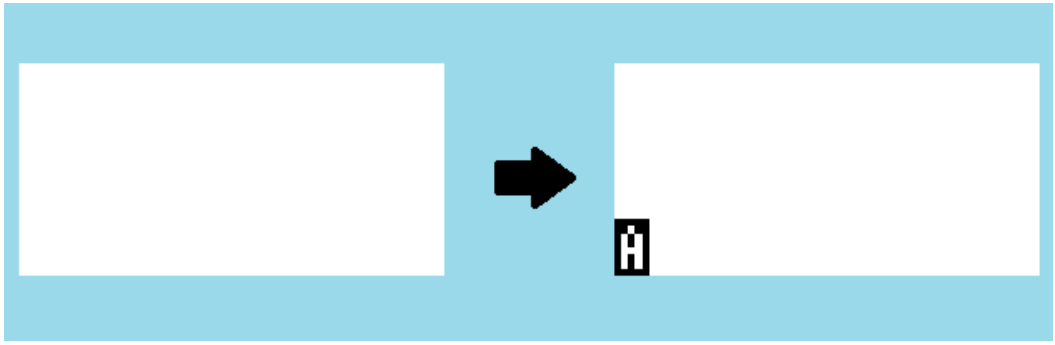


Figure 3. An image generated by the circuit for a 60 × 30 image size by ZK-optimized implementation.

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

1. 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.
2. 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]
3. Azbeg, 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]
4. 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.
5. 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.
6. 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]
7. 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.
8. 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]
9. Kang, P.; Yang, W.; Zheng, J. Blockchain private file storage-sharing method based on IPFS. *Sensors* **2022**, *22*, 5100. [CrossRef] [PubMed]
10. 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.
11. 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]
12. 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]
13. Circom Official Website. Available online: https://iden3.io/circom (accessed on 24 March 2024).
14. Snarkjs Github Repository. Available online: https://github.com/iden3/snarkjs (accessed on 4 June 2024).
15. ZCash. Available online: https://z.cash/ (accessed on 12 July 2024).
16. Ma, W.; Xiong, Q.; Shi, X.; Ma, X.; Jin, H.; Kuang, H.; Gao, M.; Zhang, Y.; Shen, H.; Hu, W. Gzkp: 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.
17. 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.
18. 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.; Schofnegger, 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/1EVrP3DzoGbmzkRmYnyEDcIQcXVU7GlOd/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.

electronics

MDPI

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: Niusha Shafiabady and Jianlong Zhou

Received: 11 June 2024
Revised: 23 July 2024
Accepted: 24 July 2024
Published: 31 July 2024

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 $\mathbf{x} \in \boldsymbol{\mathcal{X}} \subseteq \mathbb{R}^d$ and some additional protected attributes $\mathbf{s} \in \boldsymbol{\mathcal{S}} \subseteq \mathbb{R}^p$, 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 $h : \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{S}} \rightarrow \mathcal{Y}$ from a class $\mathbf{H}$. The protected attributes $\mathbf{s} \in \boldsymbol{\mathcal{S}}$ in our study are assumed to be binary with a special value n denoting the unprivileged group. For example, $\boldsymbol{\mathcal{S}}$ could be race and n "non-White"; therefore, the binary nature of $\boldsymbol{\mathcal{S}}$ 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 $\boldsymbol{\mathcal{X}}$, and protected attribute set $\boldsymbol{\mathcal{S}}$ with domain value $\{w, n\}$. The discrimination in $\mathcal{D}$ with respect to the group $\boldsymbol{\mathcal{S}} = n$ 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 $\boldsymbol{\mathcal{S}}$ [21]. FPR and TPR with respect to protected attribute $\mathbf{s} \in \boldsymbol{\mathcal{S}}$ 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

```
baselineCount ← sum of entries in 'Largest_Category' of X_train
totalCount ← number of entries in X_train
baselineProportion ← baselineCount / totalCount
balancedData ← initialize an empty dataset
categories ← list of column names in X_train starting with 'combined_category_'
for each category in categories do
    categoryData ← select entries in X_train where category = 1
    categoryData ← combine categoryData with corresponding labels from y_train
    categoryCount ← number of entries in categoryData
    targetCount ← integer part of totalCount × baselineProportion
    if categoryCount < targetCount then
        sampledData ← sample targetCount from categoryData with replacement
        balancedData ← append sampledData to balancedData
    else
        balancedData ← append categoryData to balancedData
    end if
end for
return balancedData
```

Algorithm 2 Protected-Category Proportional Sampling

```
targetSamples ← 5000
sampledBalanced ← initialize an empty data set
for each column in new_categories.columns do
    categoryRows ← select rows in new_categories where column = 1
    sampledRows ← sample targetSamples entries from categoryRows with replacement
    for each col in oneHotEncodedBalanced.columns do
        sampledRows[col] ← 0
    end for
    sampledRows[column] ← 1
    sampledBalanced ← append sampledRows to sampledBalanced
end for
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)

procedure CUSTOMSMOTE(*data*, *n_samples*)
  syntheticSamples ← zero matrix of size (*n_samples*× number of columns in *data*)
  nn ← NearestNeighbors(*n_neighbors* = 7).fit(*data*)
  neighbors ← *nn*.kneighbors(*data*, return_distance = *False*)
  for *i* ← 1 **to** *n_samples* **do**
    sampleIdx ← random integer from 0 to (number of rows in *data* − 1)
    nnIdx ← random choice from *neighbors*[*sampleIdx*, 1 :]
    diff ← *data*[*nnIdx*] − *data*[*sampleIdx*]
    weight ← random number from uniform distribution between 0 and 1
    syntheticSamples[*i*] ← *data*[*sampleIdx*] + *weight* × *diff*
  end for
  return *syntheticSamples*
end procedure

Algorithm 4 Generate Balanced Synthetic Labels

procedure GENBALSYNTHLABELS(*n_samplesNeeded*, *balanceRatio*)
  *nClass*1 ← int(*n_samplesNeeded* × *balanceRatio*)
  *nClass*0 ← *n_samplesNeeded* − *nClass*1
  syntheticLabels ← [0] × *nClass*0 + [1] × *nClass*1
  SHUFFLE(*syntheticLabels*) ▷ Randomly shuffle the labels
  return *syntheticLabels*
end procedure

Algorithm 5 Protected-category SMOTE

balancedDataList ← initialize an empty list
for each *category* in *categories* **do**
  categorySubset ← select rows in *train_data* s.t. 'combined_category' == *category*
  features ← remove 'class', 'combined_category' from *categorySubset*
  nSamples ← *max_size*− number of rows in *categorySubset*
  if *nSamples* > 0 **then**
    syntheticFeatures ← PCSmote(*features*, *nSamples*)
    syntheticLabels ← GenBalSynthLabels(*nSamples*, *balanceRatio*)
    syntheticFeatures['*class*'] ← *syntheticLabels*
    syntheticFeatures['*combined_category*'] ← *category*
    categorySubsetBalanced ← concatenate *categorySubset* and *syntheticFeatures*
  else
    categorySubsetBalanced ← *categorySubset*
  end if
  append *categorySubsetBalanced* to *balancedDataList*
end for
balancedData ← 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

procedure PCADASYNCATEGORIES(*data*, *labels*, *n_samplesNeeded*, *n_neighbors*)
  $n\_neighbors \leftarrow n\_neighbors + 1$ ▷ Including the data point itself
  $nn \leftarrow \text{NearestNeighbors}(n\_neighbors).\text{fit}(data)$
  $distances, indices \leftarrow nn.\text{kneighbors}(data)$
  $densities \leftarrow$ zero array of length$(data)$
  for $i \leftarrow 0$ **to** length$(data) - 1$ **do**
    $current\_category \leftarrow labels[i]$
    $neighbor\_indices \leftarrow indices[i][1:]$ ▷ Skip the self index
    $densities[i] \leftarrow \text{SUM}(labels[neighbor\_indices] == current\_category)$
  end for
  $weights \leftarrow 1/(densities + 1)$ ▷ Add 1 to prevent division by zero
  $weights \leftarrow weights/\text{SUM}(weights)$ ▷ Normalize weights
  $syntheticSamples \leftarrow$ empty list
  $sampleIndices \leftarrow$ random choice with replacement from length$(data)$ using $weights$
  for each idx **in** $sampleIndices$ **do**
    $baseIdx \leftarrow idx$
    $neighborIdx \leftarrow \text{RANDOMCHOICE}(indices[baseIdx][1:])$
    $diff \leftarrow data[neighborIdx] - data[baseIdx]$
    $syntheticSample \leftarrow data[baseIdx] + \text{RANDOM}() \times diff$
    append $syntheticSample$ to $syntheticSamples$
  end for
  return array$(syntheticSamples)$
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

balancedDataList ← initialize an empty list
for each *category* in *categories* **do**
    categorySubset ← select from *new_data*3 s.t. 'combined_category' == *category*
    features ← remove 'class', 'combined_category' from *categorySubset*
    categoryLabels ← extract 'combined_category' from *categorySubset*
    nSamplesNeeded ← *max_size* minus number of rows in *categorySubset*
    if *nSamplesNeeded* > 0 **then**
        syntheticFeatures ← PCAdasyn(*features*, *categoryLabels*, *nSamplesNeeded*)
        syntheticLabels ← GenBalSynthLabels(*nSamplesNeeded*, *balanceRatio*)
        syntheticFeatures['*class*'] ← *syntheticLabels*
        syntheticFeatures['*combined_category*'] ← *category*
        categorySubsetBalanced ← concatenate *categorySubset* with *syntheticFeatures*
    else
        categorySubsetBalanced ← *categorySubset*
    end if
    append *categorySubsetBalanced* to *balancedDataList*
end for
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.

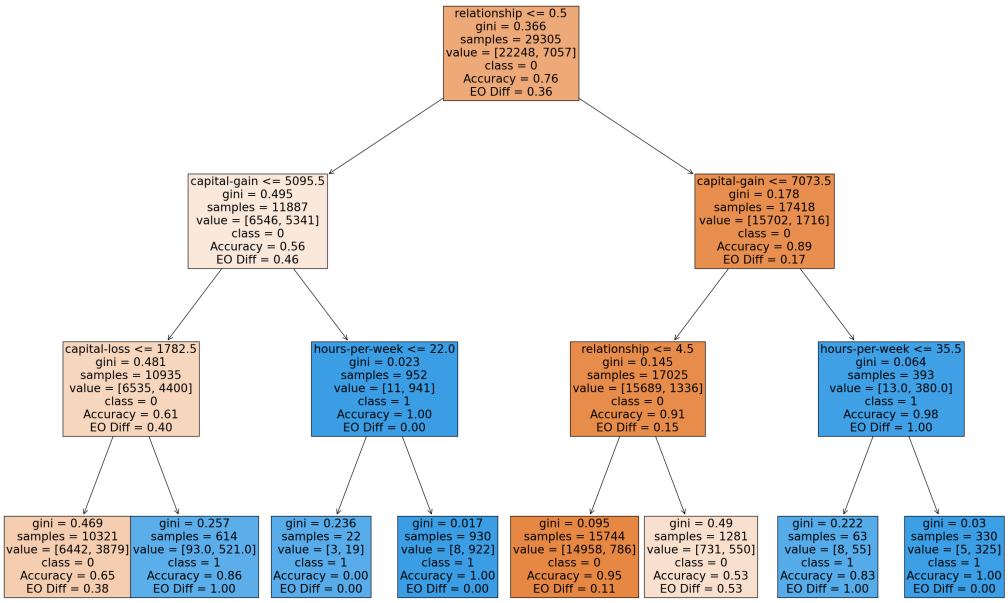


Figure 1. Example decision tree trained on Adult Income with no sampling.

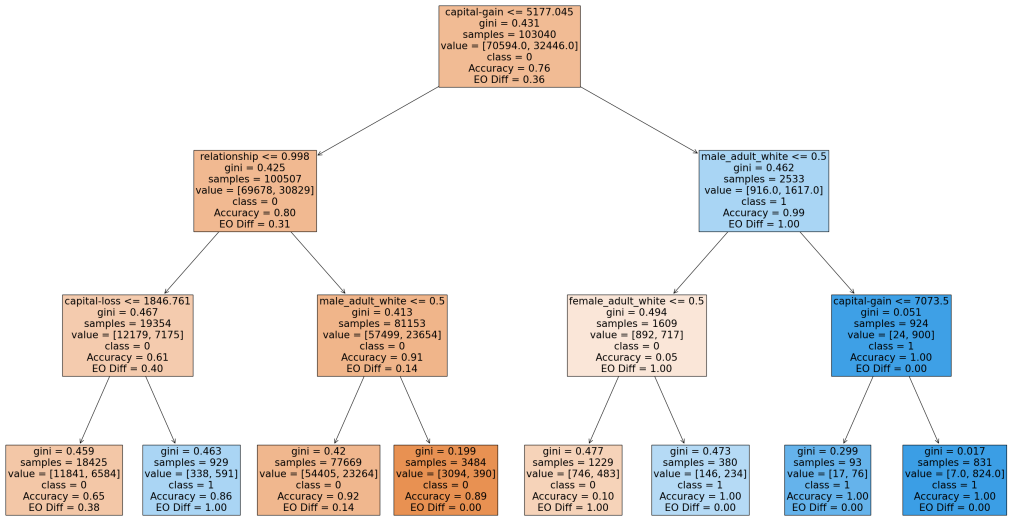


Figure 2. Decision tree of PC-ADASYN on adult income.

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.

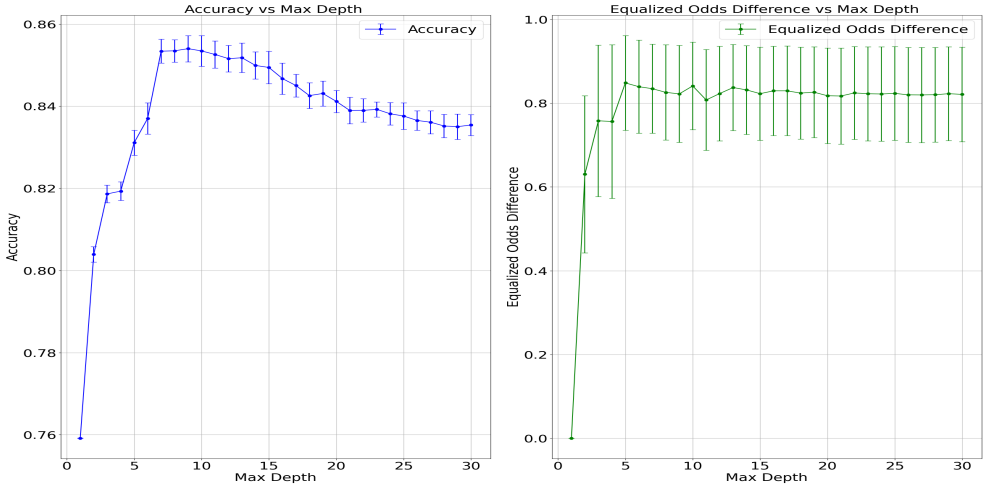


Figure 3. Plots of Adult Income using no sampling, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

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 oversampling 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.

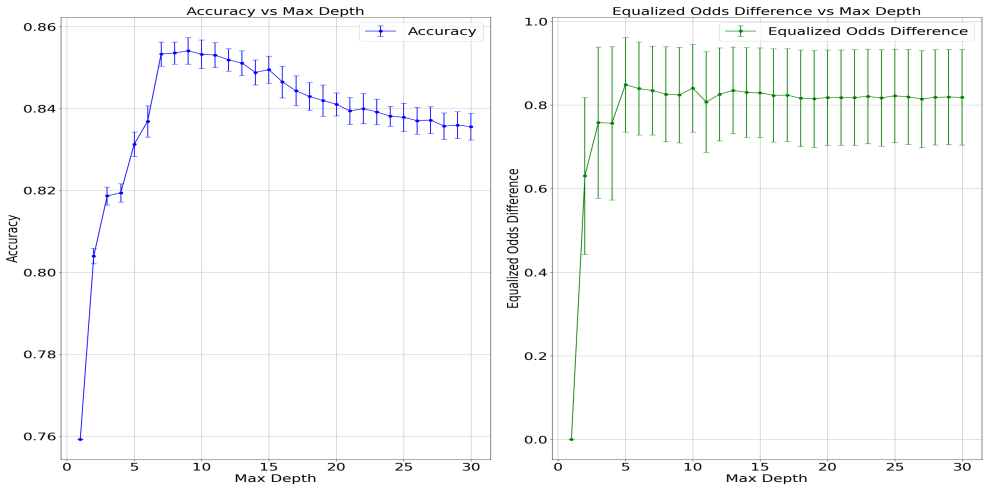


Figure 4. Plots of Adult Income using oversampling, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

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.

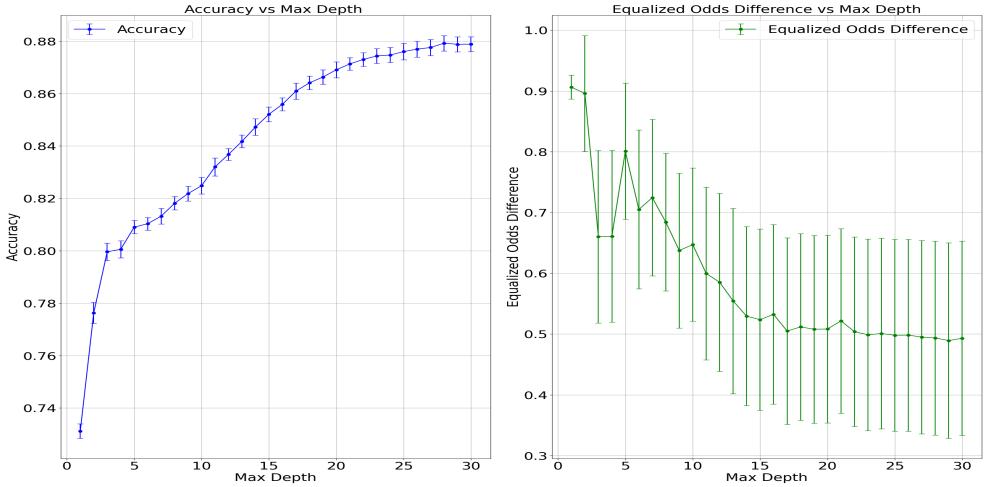


Figure 5. Plots of Adult Income using proportional sampling, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

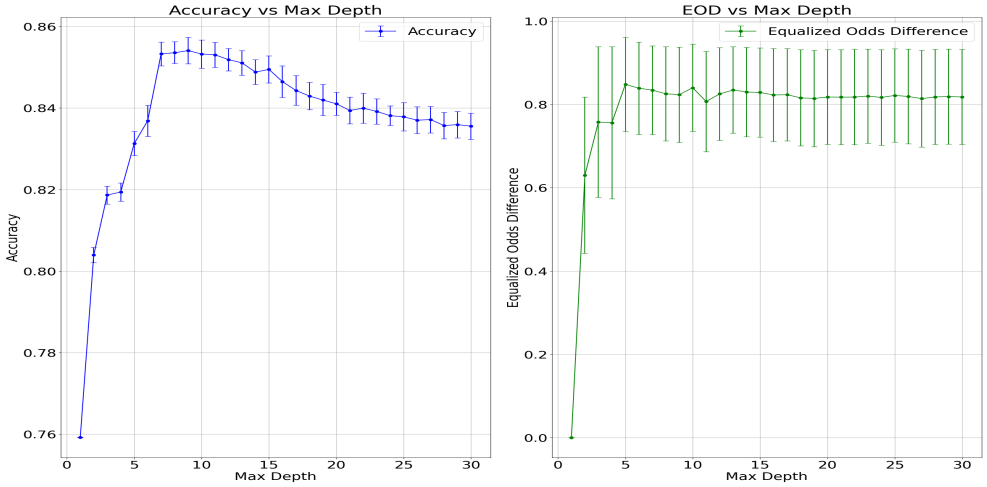


Figure 6. Plots of Adult Income using PC-SMOTE, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

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.

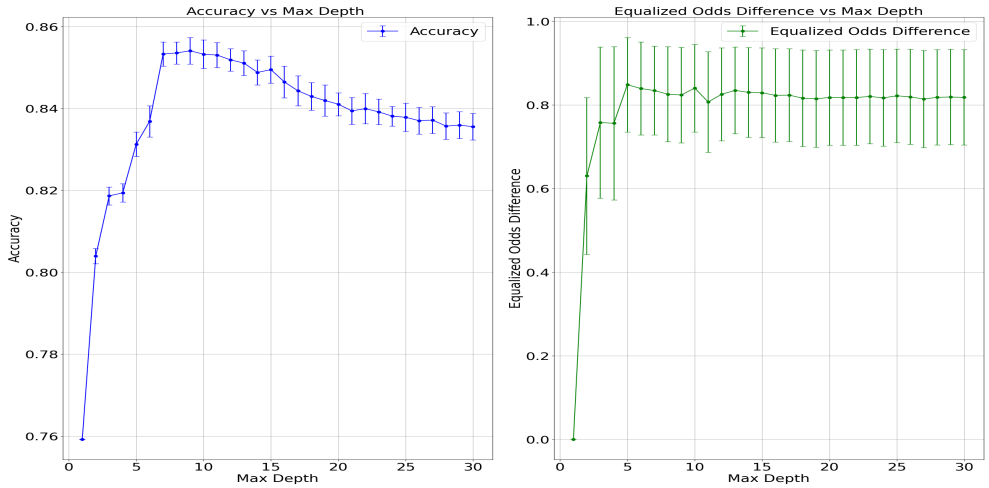


Figure 7. Plots of Adult Income using PC-ADASYN, showing accuracy and EOD with 95% confidence intervals against the maximum tree depth ranging from 1 to 30.

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.; Cemballi, 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.; Žliobaitè, 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 Massachusets, 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, ICDSST 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.

electronics

MDPI

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 [1]

[1] Department of Technology Management for Innovation, School of Engineering, The University of Tokyo, Tokyo 113-8656, Japan
[2] 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

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 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].

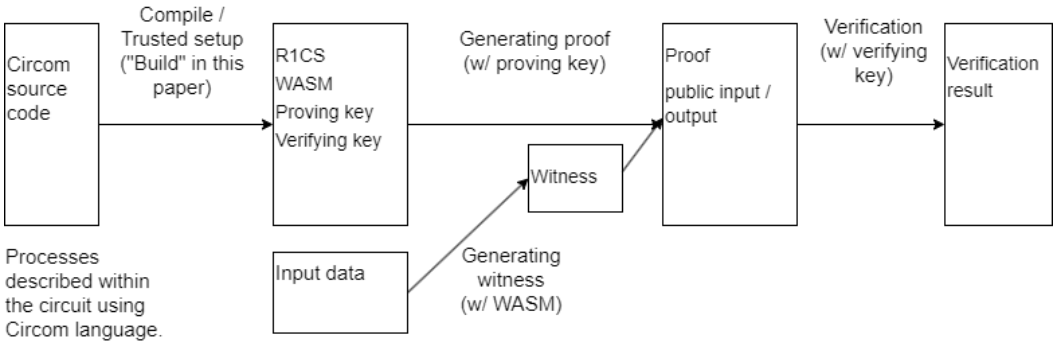


Figure 1. Zero-knowledge proof using circom.

4. Proposed System

Figure 2 presents an overview of the proposed system.

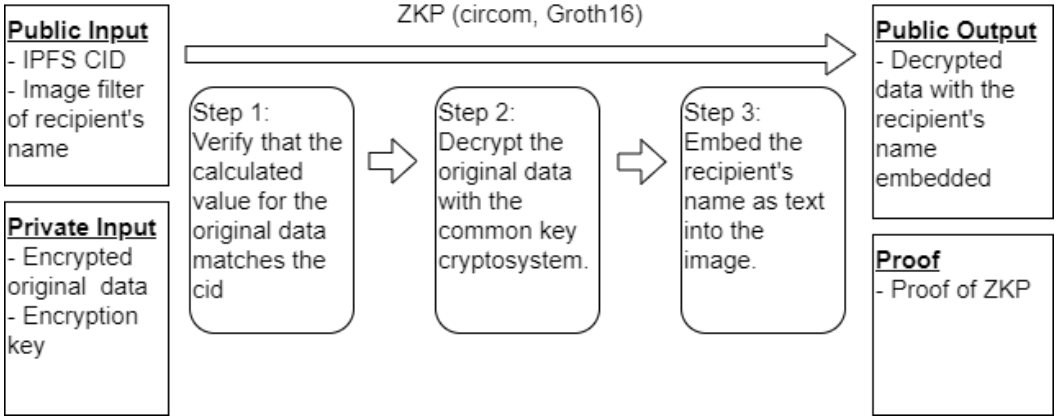


Figure 2. Process flow of ZKP: an example of image data processing.

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.

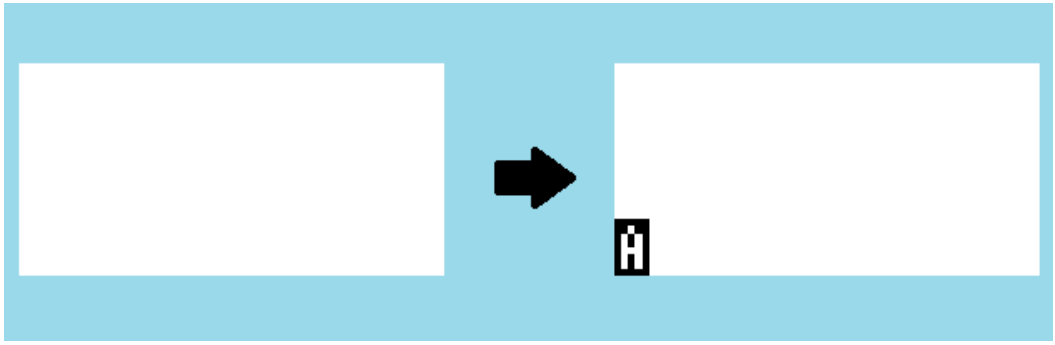


Figure 3. An image generated by the circuit for a 60 × 30 image size by ZK-optimized implementation.

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

1. 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.
2. 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]
3. Azbeg, 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]
4. 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.
5. 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.
6. 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]
7. 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.
8. 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]
9. Kang, P.; Yang, W.; Zheng, J. Blockchain private file storage-sharing method based on IPFS. *Sensors* **2022**, *22*, 5100. [CrossRef] [PubMed]
10. 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.
11. 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]
12. 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]
13. Circom Official Website. Available online: https://iden3.io/circom (accessed on 24 March 2024).
14. Snarkjs Github Repository. Available online: https://github.com/iden3/snarkjs (accessed on 4 June 2024).
15. ZCash. Available online: https://z.cash/ (accessed on 12 July 2024).
16. Ma, W.; Xiong, Q.; Shi, X.; Ma, X.; Jin, H.; Kuang, H.; Gao, M.; Zhang, Y.; Shen, H.; Hu, W. Gzkp: 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.
17. 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.
18. 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.; Schofnegger, 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/1EVrP3DzoGbmzkRmYnyEDcIQcXVU7GlOd/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.

electronics

MDPI

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

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.

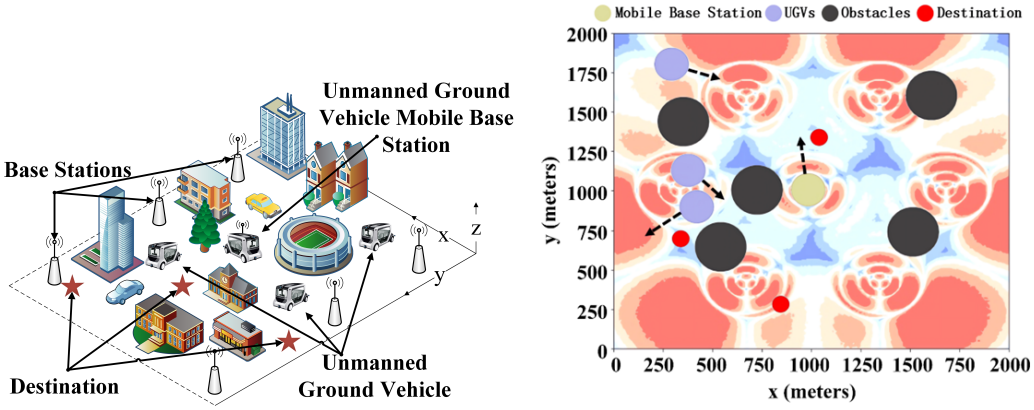
2.1. Problem Description

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.



(**a**) (**b**)

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:

$$\mathrm{SIR}_t \triangleq \frac{p_t^{I_t}}{\sum_{m \neq I_t} p_t^m} \tag{1}$$

where $I_t \in \{1, \cdots, M\}$ 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:

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

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:

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

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:

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

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.

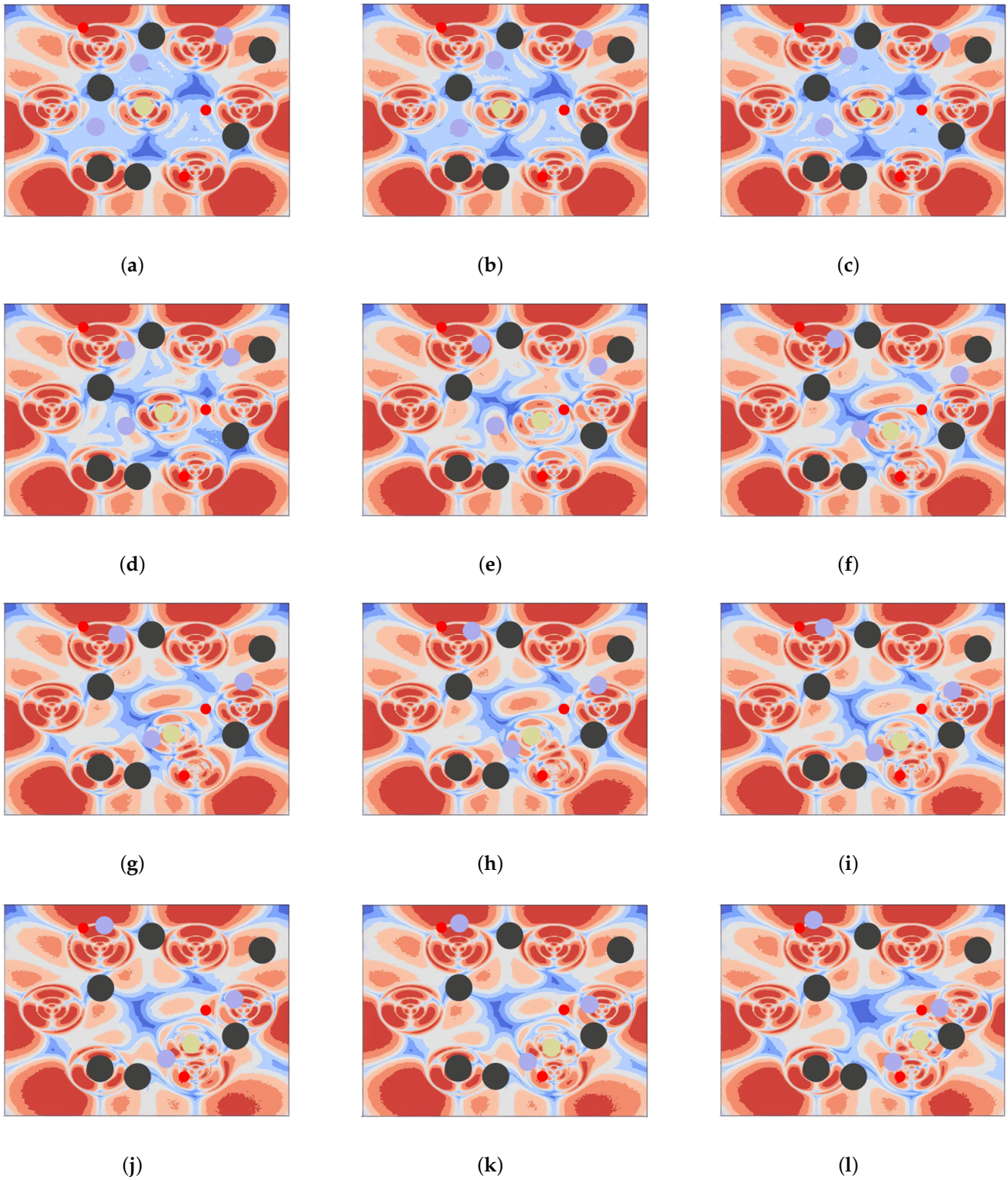


Figure 2. The results of changing communication in the environment as the mobile BS moves. The red areas indicate better communication, whereas the blue areas indicate poorer communication, Brown circles represent the mobile BS, blue circles represent UGVs, black circles represent obstacles, and red 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 $s = (s_1, s_2, \ldots, s_N)$. For each UGV u, the state is defined as $s_u = (s_{Pu}, s_{Eu})$, where $s_{Pu} = \left(x_u, y_u, v_{xu}, v_{yu}, SIR_u\right)$ is a combination of position (x_u, y_u), speed (v_{xu}, v_{yu}), and SIR_u. Additionally, $s_{Eu} = \left(x_{ug}, y_{ug}, x_0, y_0, v_{ox}, v_{oy}\right)$ 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, $s_{Eu} = \left(x_{ug}, y_{ug}, x_0, y_0, v_{ox}, v_{oy}\right)$ 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 $a = (a_1, a_2, \ldots, a_N)$, 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:

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

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 $r(s_t, a_t)$.

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

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:

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

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:

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

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:

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

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

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

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

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

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.

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

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:

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

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

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

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

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

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].

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

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 $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_N\}$. The term $\boldsymbol{\pi} = \{\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_N\}$ 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:

$$\begin{aligned} \nabla_{\theta_i} J(\theta_i) = & \mathbb{E}_{s\sim p^{\boldsymbol{\mu}}, a_i \sim \boldsymbol{\pi}_i} [\nabla_{\theta_i} \log \boldsymbol{\pi}_i(a_i \mid o_i) \\ & Q_i^{\boldsymbol{\pi}}(\mathbf{x}, a_1, \ldots, a_N)] \end{aligned} \tag{16}$$

In our setting, the total actions $a_1, \ldots, a_N$ are fed into $Q_i^{\boldsymbol{\pi}}(\mathbf{x}, a_1, \ldots, a_N)$, 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, $(o_1, \ldots, o_N)$, 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:

$$\begin{aligned} \nabla_{\theta_i} J(\boldsymbol{\mu}_i) = & \mathbb{E}_{\mathbf{x}, a\sim \mathcal{D}} [\nabla_{\theta_i} \boldsymbol{\mu}_i(a_i \mid o_i) \\ & \nabla_{a_i} Q_i^{\boldsymbol{\mu}}(\mathbf{x}, a_1, \ldots, a_N)|_{a_i = \boldsymbol{\mu}_i(o_i)}] \end{aligned} \tag{17}$$

where the transitions $(\mathbf{x}, \mathbf{x}', a_1, \ldots, a_N, r_1, \ldots, r_N)$ 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:

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

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.

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

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:

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

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 nth 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:

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

The actor is updated using the sampled policy gradient:

$$\begin{aligned} \nabla_{\theta_i} J \approx & \frac{1}{S} \sum_{j} \nabla_{\theta_i} \boldsymbol{\mu}_i(o_i^j) \\ & \nabla_{a_i} Q_i^{\mu}(\mathbf{x}^j, a_1^j, \ldots, a_i, \ldots, a_N^j)|_{a_i = \boldsymbol{\mu}_i(o_i^j)} \end{aligned} \tag{22}$$

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

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

where

$$\boldsymbol{\mu}' = \left\{ \boldsymbol{\mu}_{\theta'_1}, \ldots, \boldsymbol{\mu}_{\theta'_N} \right\} \tag{24}$$

is the collection of goal policies with postponed parameters $\boldsymbol{\theta}_i$.

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 $\mathcal{D}$.
$E_{count} = E_{count} + 1$
if $E_{count} \geq$update episode **then**
for $g = 1$ to critic updates steps **do**
Sample batch $\mathcal{B}$ from $\mathcal{D}$
Set $y^j = r_i^j + \gamma Q_i^{\boldsymbol{\mu}'}(\mathbf{x}'^j, a'_1, \ldots, a'_N)\Big|_{a'_k = \boldsymbol{\mu}'_k\left(o_k^j\right)}$
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

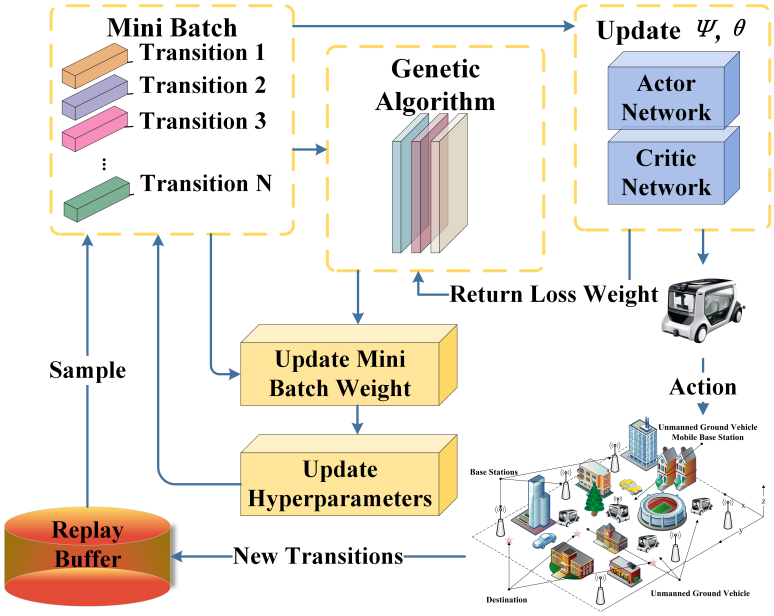


Figure 3. Detailed diagram of the GA-MADDPG algorithm.

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 × 2 km² 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 dBm, the communication interruption threshold is set to γ_{th} = 0 dB, and the noise power is defined as $\sigma^2 = -65$ 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

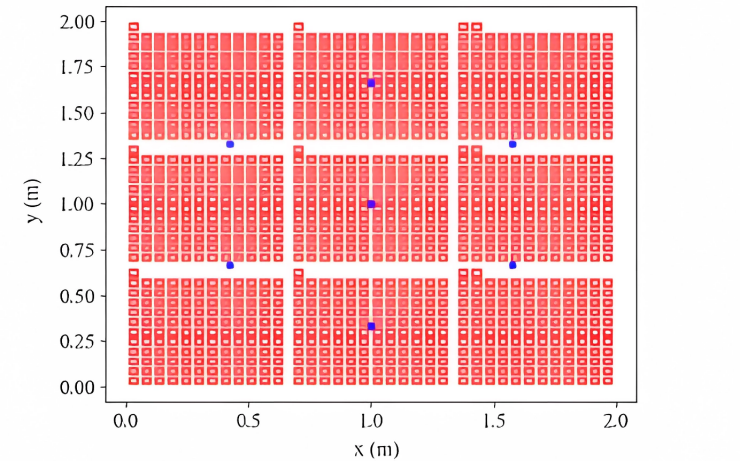
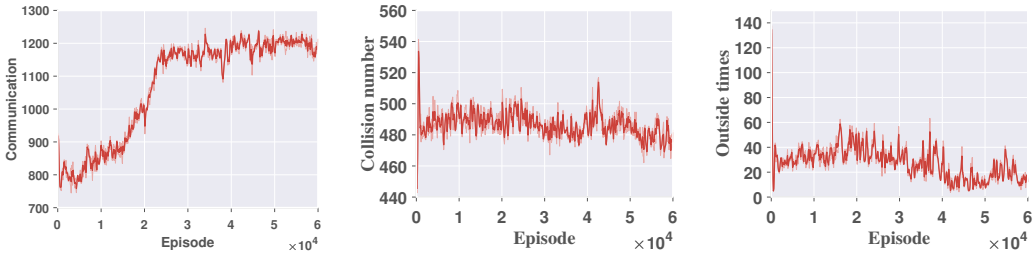


Figure 4. Plan view of base station model distribution.

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.



(**a**) (**b**) (**c**)

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 RMATD3 [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.

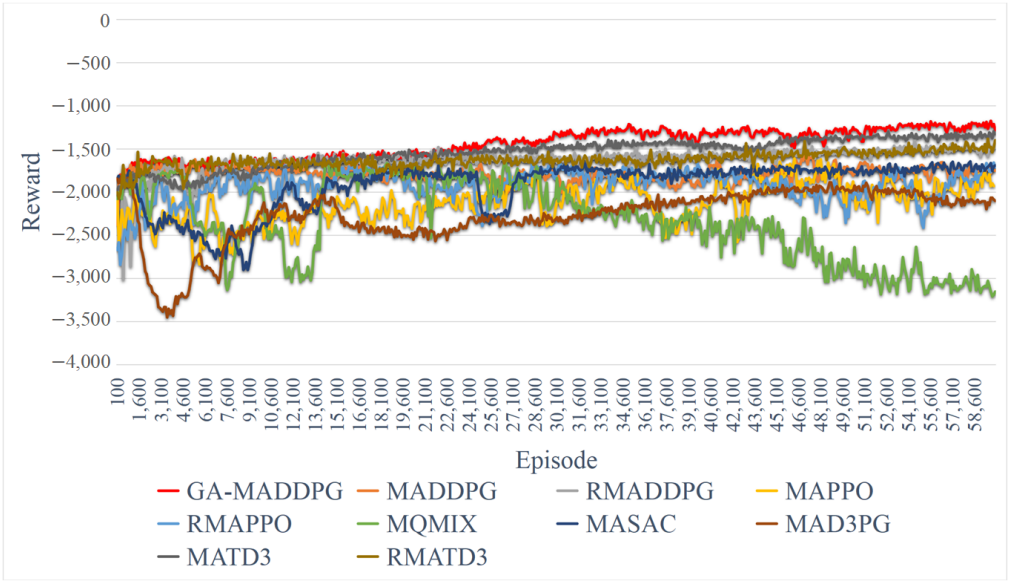


Figure 6. Average cost of the GA-MADDPG and other advanced algorithms.

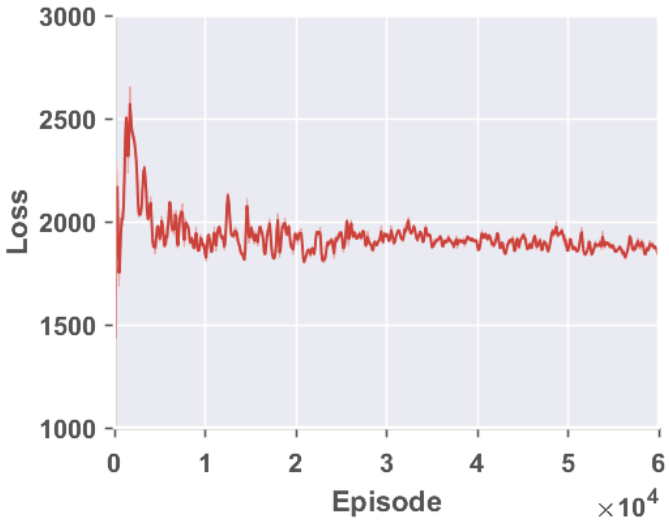


Figure 7. Evolution of loss function.

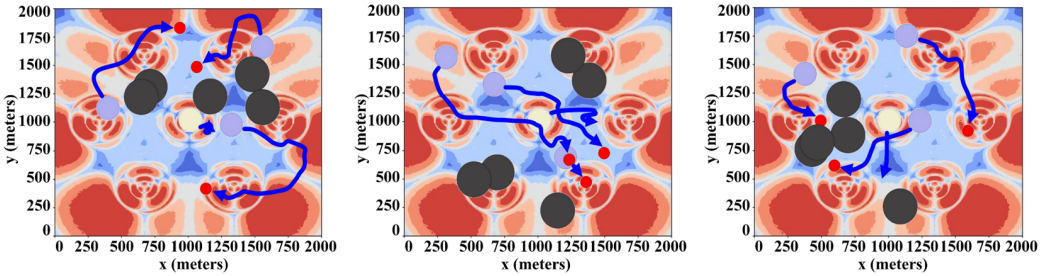


Figure 8. Some UGV path maps based on GA-MADDPG.

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 difficultly 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.

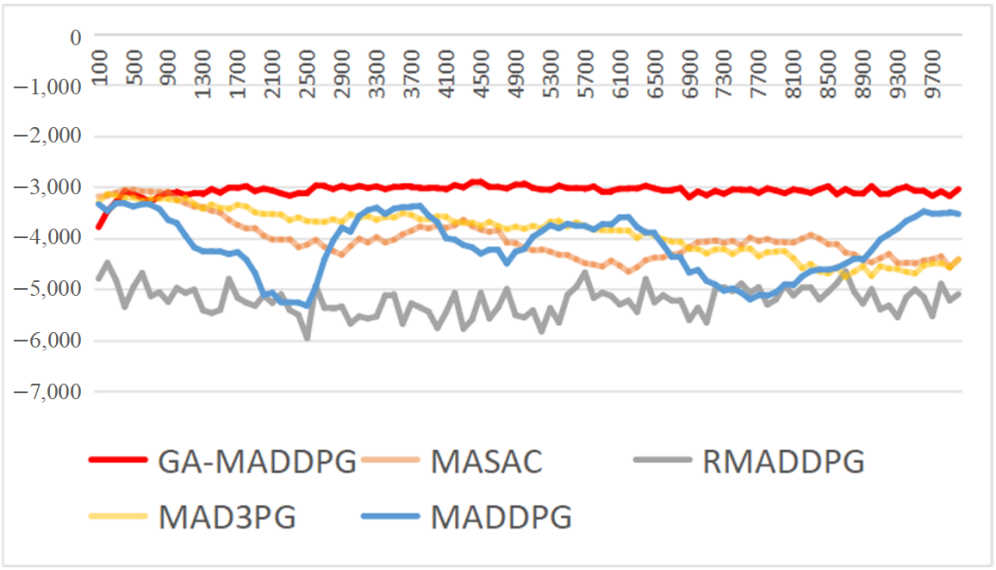


Figure 9. Average cost comparison of generalization environment 1.

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.

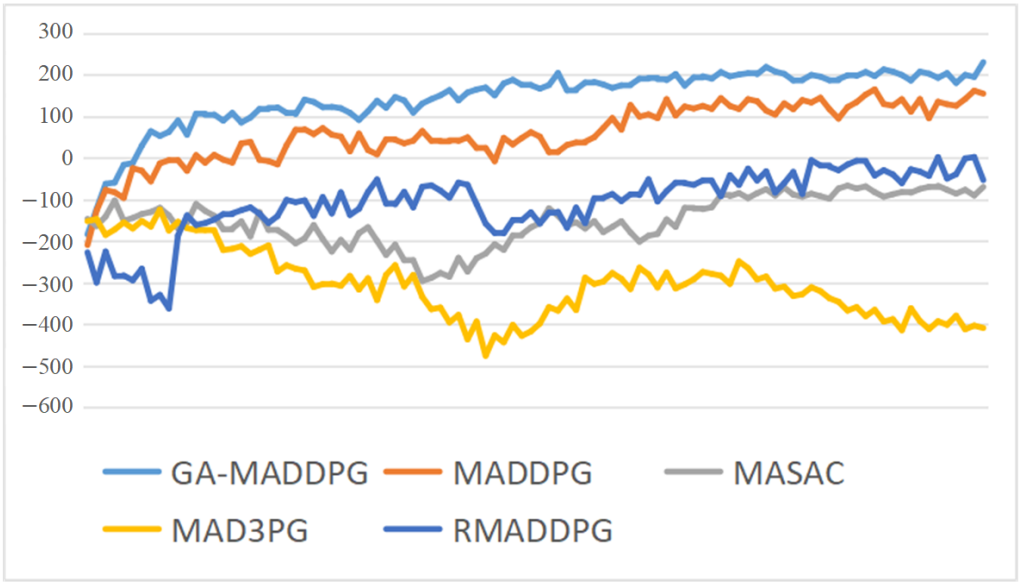


Figure 10. Average cost comparison of generalization environment 2.

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.

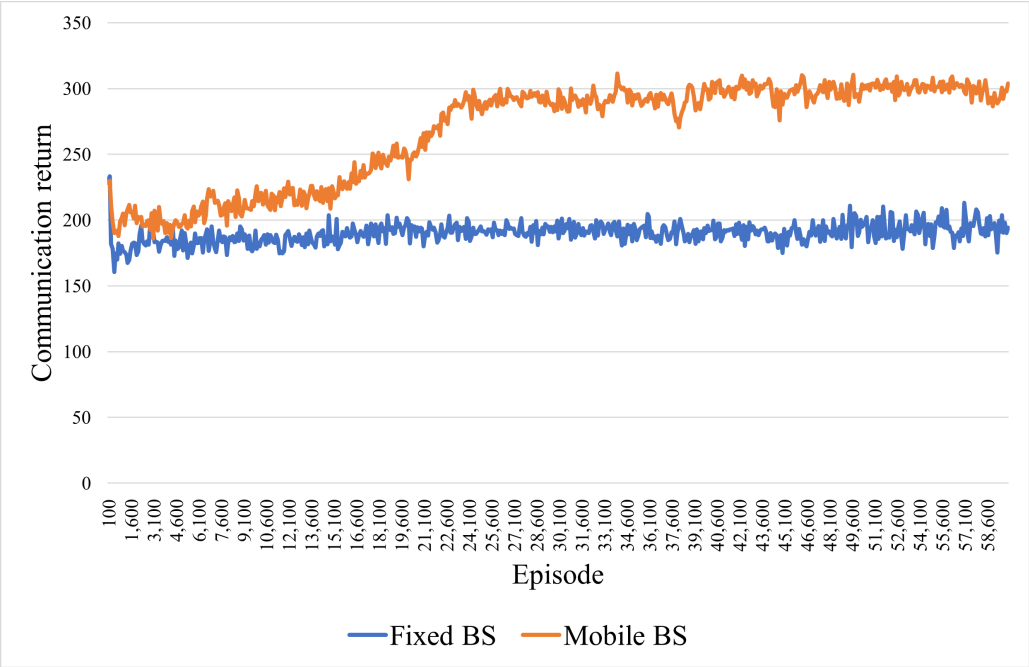


Figure 11. Comparison of communication returns between mobile BS and fixed BS.

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.; Vinitsky, 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]

electronics

MDPI

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

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.

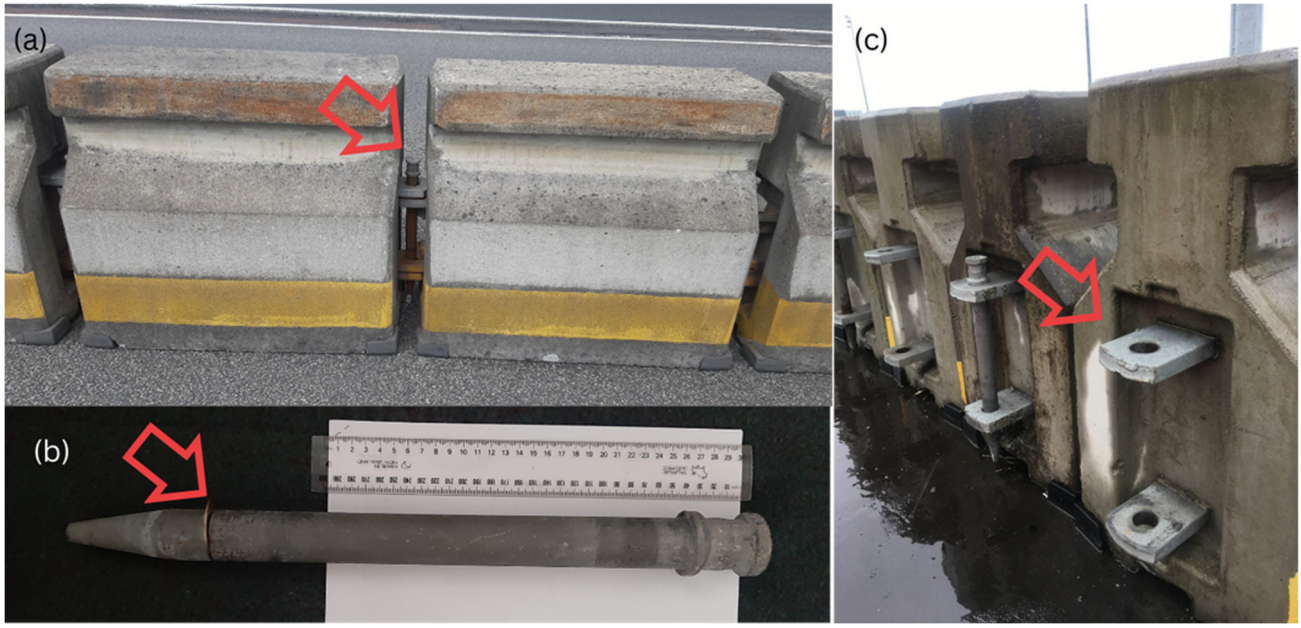


Figure 1. A movable concrete barrier system and its safety challenges. (**a**) Pin position requiring fixing, (**b**) metal pin with detachable safety ring, and (**c**) movable concrete barrier joints without metal pins.

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).

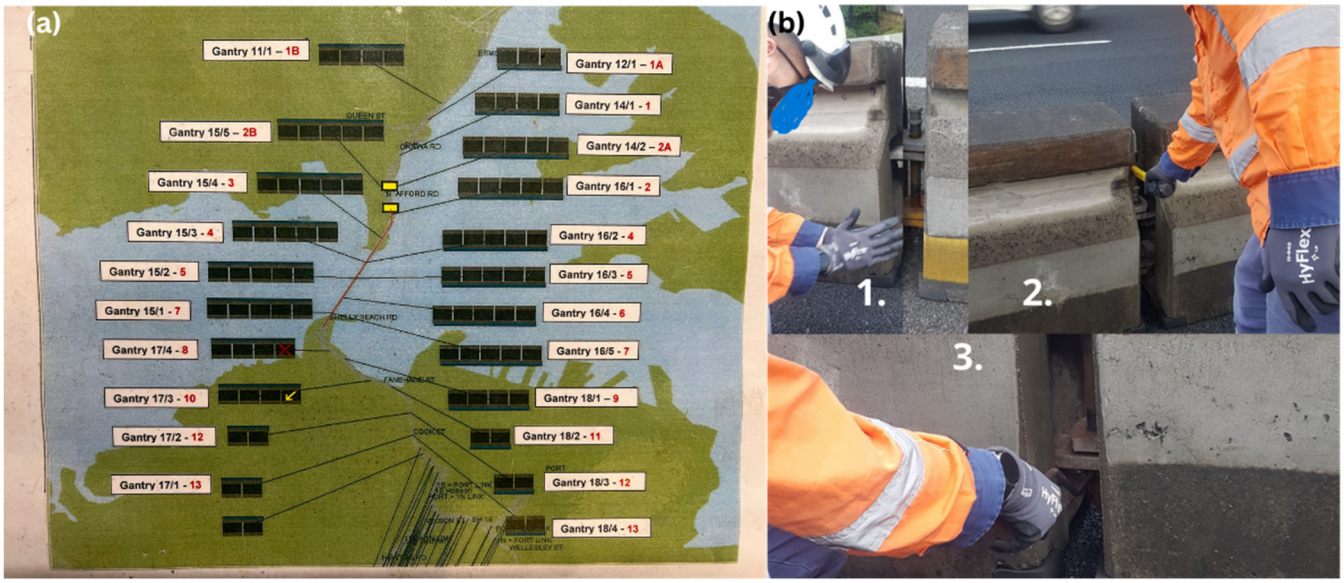


Figure 2. The scale of the challenges for manual inspection and fixing of the metal pin: (**a**) various locations of movable concrete barriers on and around the Auckland Harbour Bridge; (**b**) finding an unsafe pin and fixing it manually.

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.

$$\mathrm{Y} = 0.299\mathrm{R} + 0.587\mathrm{G} + 0.114\mathrm{B} \tag{1}$$

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:

$$\mathrm{x_c} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{x_i} \cdot \mathrm{w_i}}{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{w_i}} \tag{2}$$

where $\mathrm{x_c}$ is the centroid location, $\mathrm{x_i}$ is the pixel location, and $\mathrm{w_i}$ is the pixel intensity. Multiplying $\mathrm{x_i}$ by $\mathrm{w_i}$ 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).

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

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 $\sum_{k,t}$ 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 $\mid F_t - F_t - 1 \mid$ to identify changes and optical flow, represented by $\nabla l_t \cdot v_t$, to analyse motion patterns and speeds, with α adjusting their relative contributions.

$$\Delta M_t = \mid F_t - F_t - 1 \mid + \alpha \cdot \nabla l_t \cdot v_t \quad (4)$$

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)).

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(z_k - H_k \hat{x}_{k|k-1}\right) \quad (5)$$

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].

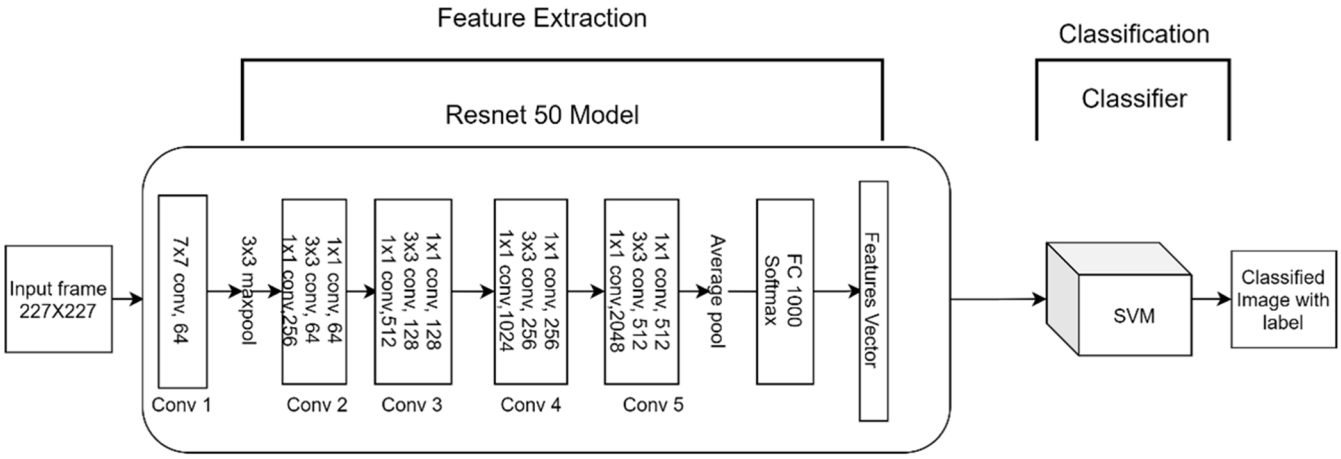


Figure 3. Integration of transfer learning with traditional ML: ResNet 50 extracts features from input images, which an SVM classifies. Integrating ResNet 50's deep learning capabilities to generate feature space without relying on expert knowledge combined with a traditional classifier (e.g., SVM) presents 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.

$$\hat{y} = \text{softmax}(W f_{RNN}(f_{CNN}(X)) + b) \tag{6}$$

where the loss function used in transfer learning is

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, f(X_i; \theta)) \tag{7}$$

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.

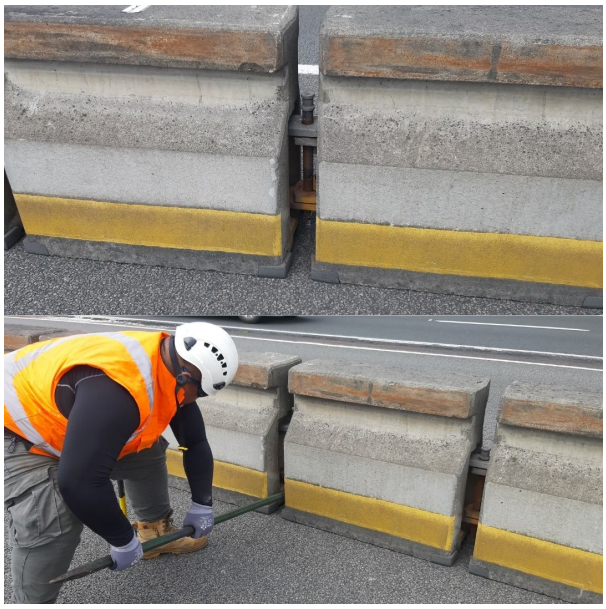


Figure 5. Pin manually pushed out of place by NZTA staff at the author's request. The process was challenging and labour-intensive, and the staff member needed several minutes per pin for adjustments. Note: Such an option was not viable as it does not represent the natural environment of the ROI where 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.

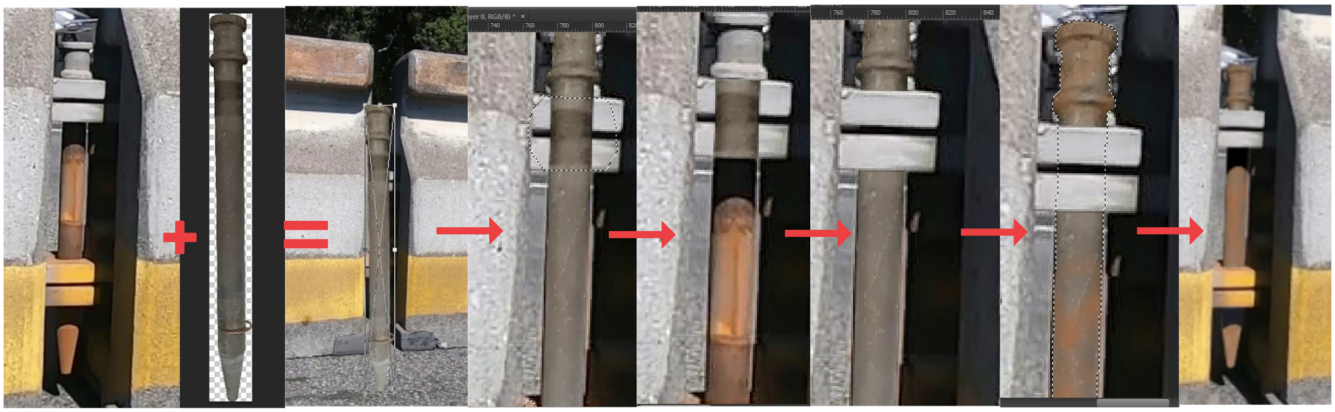


Figure 6. Synthetic data generation: The image illustrates the various stages of manipulating a video frame to create synthetic images depicting a metal pin in unsafe positions. Starting with a standalone image of the metal pin, the process involves adjusting its orientation, position, and environmental 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.

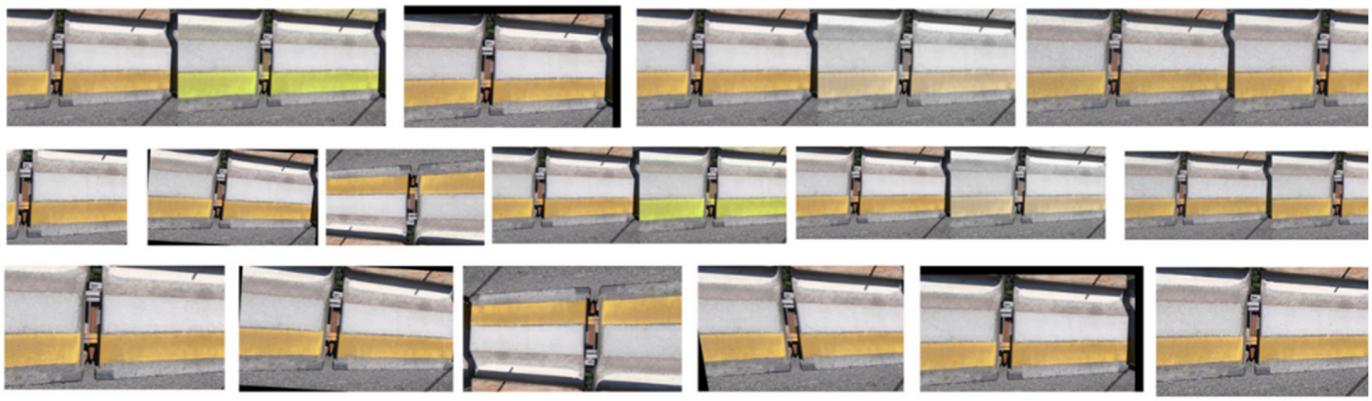


Figure 7. Illustrative outputs from data augmentation techniques showcase the range of image transformations applied to enhance model training. It includes geometric transformations such as rotation and scaling, affine transformations with various translations and shearing, and visual effects such as 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.

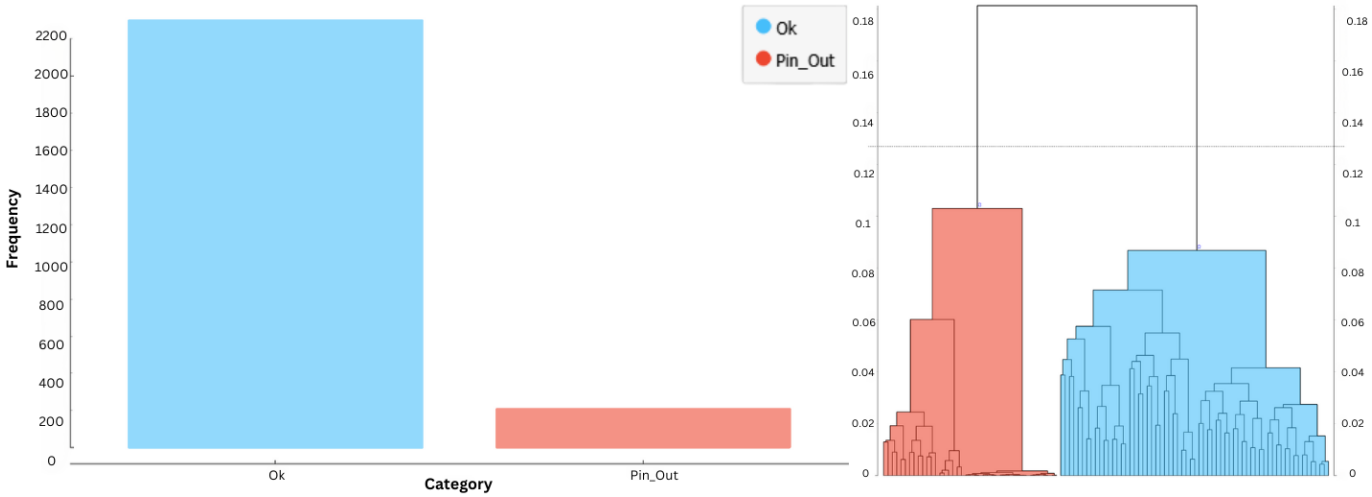


Figure 8. The dendrogram graph, derived from the data shown by the bar graph, shows two clusters (Pin_OK (red) and Pin_Out (blue)) and a visual separation of the generated multidimensional feature space.

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%]
Neural Network		PIN_OK	[98.9%	1.1%
		PIN_OUT	0	100%]
SVM		PIN_OK	[98.1%	1.9%
		PIN_OUT	0	100%]

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:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{8}$$

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:

$$\text{Accuracy} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9}$$

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.

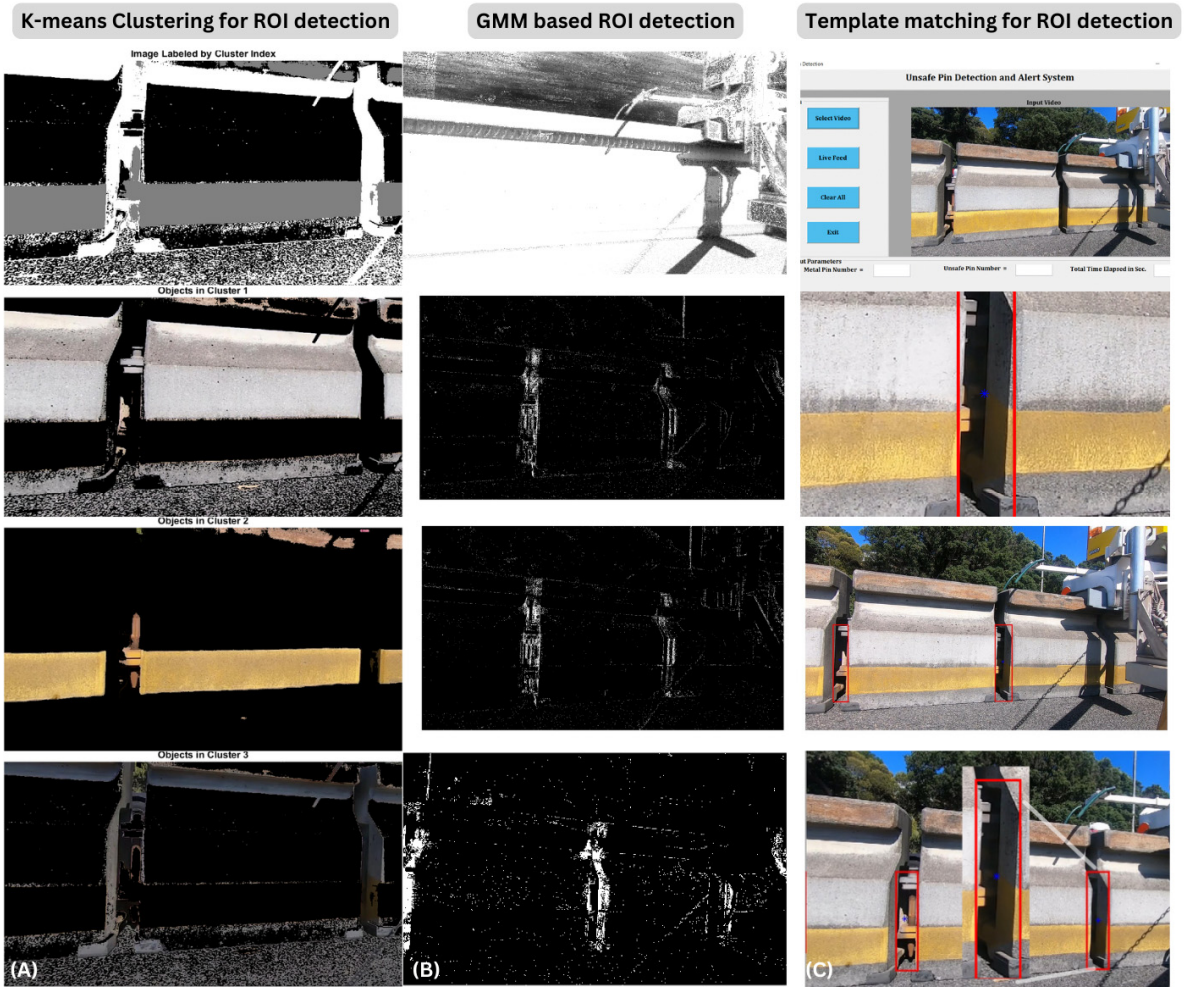


Figure 9. Comparative analysis of promising ROI detection techniques in automated pin detection: (1) colour-based segmentation using K-Means clustering in HSV colour space, highlighting the unique

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.

$$\text{Area} = \sum_{(i,j)\in R} 1 \tag{10}$$

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:

$$\text{Centroid}_x = \frac{1}{N}\sum_{(i,j)\in R} i,\ \text{Centroid}_y = \frac{1}{N}\sum_{(i,j)\in R} j \tag{11}$$

Bounding box: The bounding box, which encloses the region, is defined by the minimum and maximum x and y coordinates of the pixels.

$$\text{BoundingBox} = [x_{min}, y_{min}, x_{max} - x_{min} + 1, y_{max} - y_{min} + 1] \tag{12}$$

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.

$$\text{MajorAxisLength} = 2\sqrt{\frac{\lambda_1}{N}},\ \text{MinorAxisLength} = 2\sqrt{\frac{\lambda_2}{N}} \tag{13}$$

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.

$$\text{Orientation} = \frac{1}{2}\arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \tag{14}$$

Perimeter: The perimeter is measured by summing the distances between consecutive boundary pixels.

$$\text{Perimeter} = \sum_{\text{boundrypixels}} \text{distance between consecutive boundary pixels} \tag{15}$$

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.

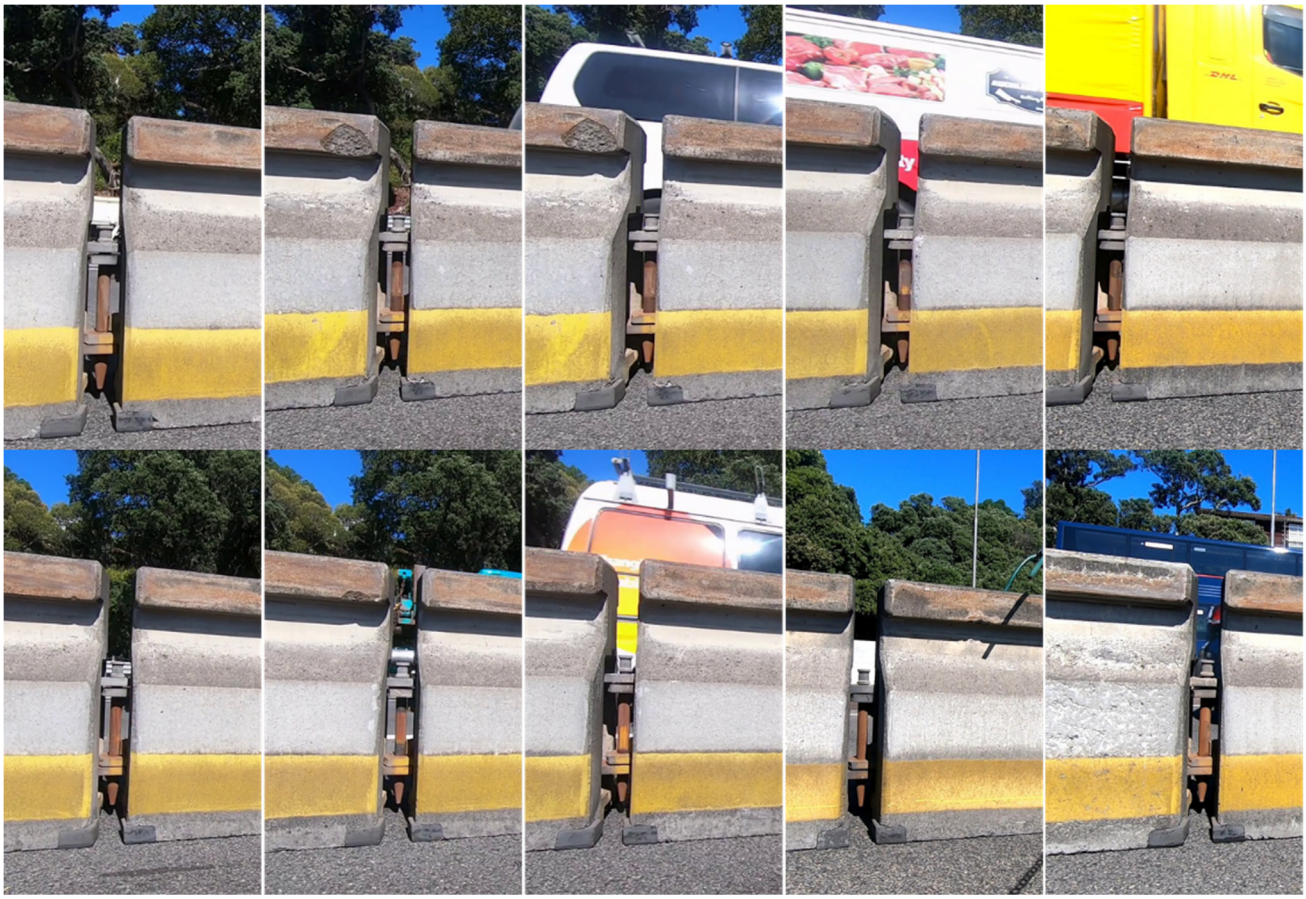


Figure 10. Illustration of the background challenges in ROI detection.

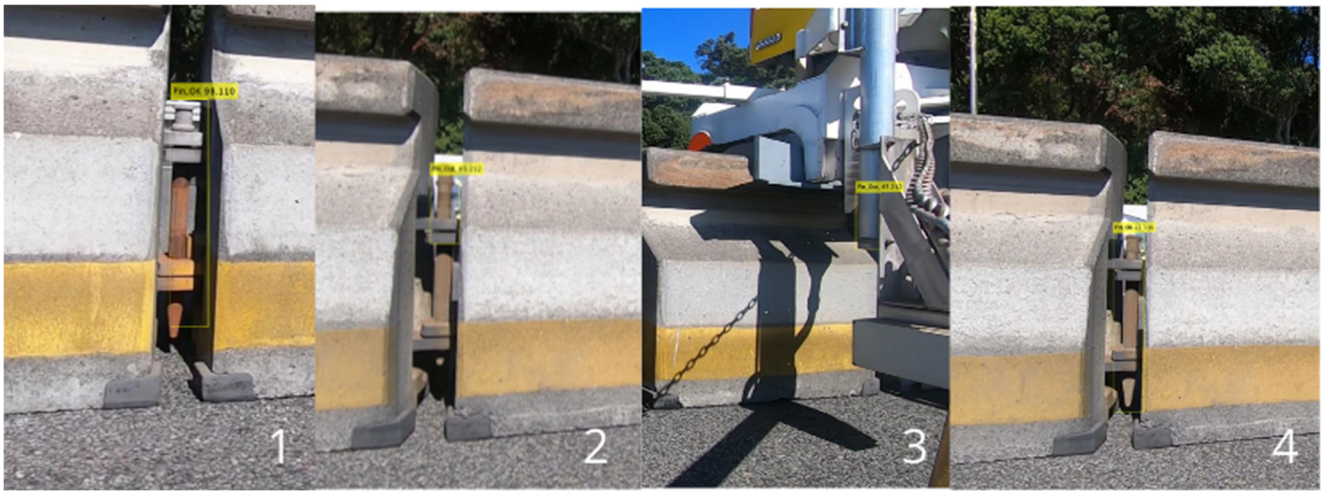


Figure 11. Pin detection examples mixed with false-positive detection (3) illustrate a high level of background noise that looks similar to features of the ROI.

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.

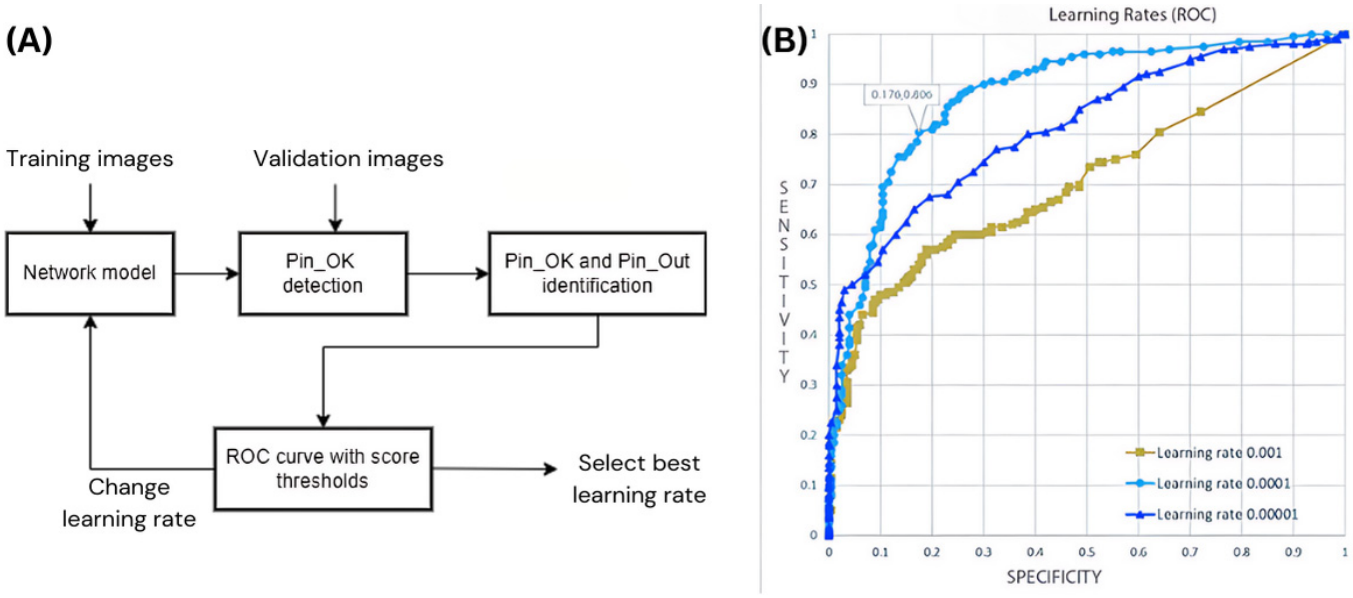


Figure 12. The training process to find the best network model based on learning rate: (**A**) flow of the training process and (**B**) ROC curves for different learning rates showing the performance of learning rates. The learning rate of 0.0001 was chosen as it balances training speed and accuracy, 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

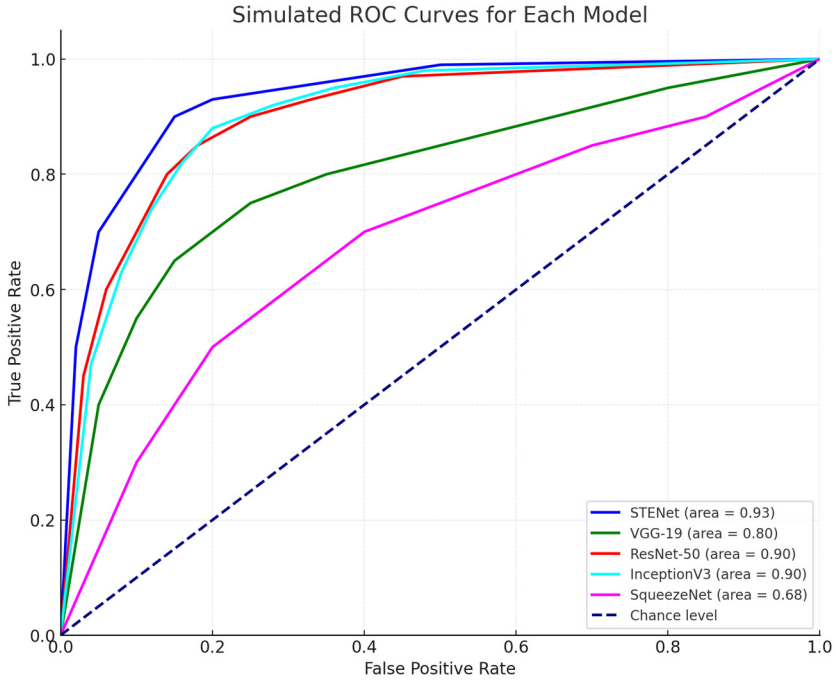


Figure 13. ROC curve comparison of Table 8.

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).

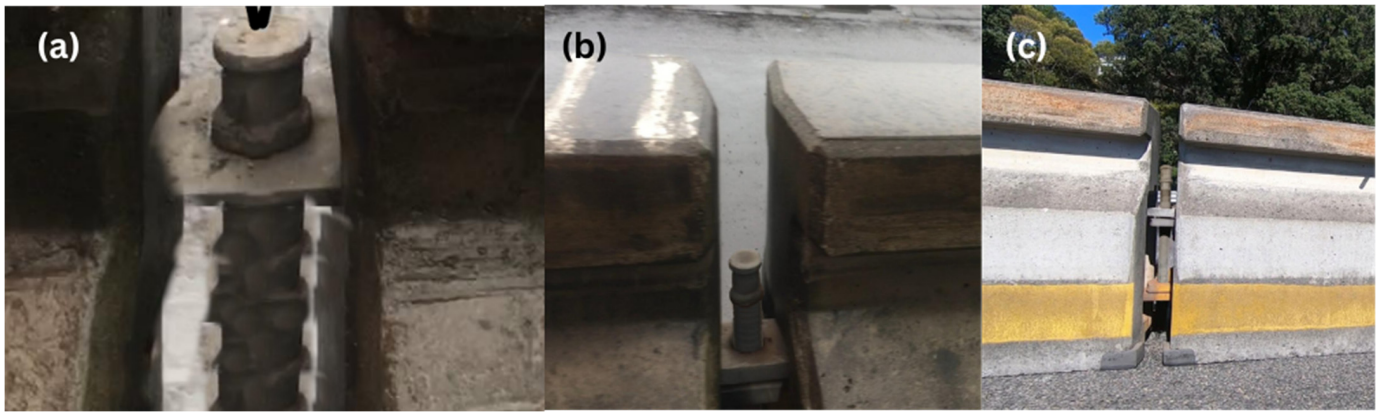


Figure 14. Cloning process to create synthetic frames depicting the 'PIN OUT' position. (**a**) Failed attempts until success (**b**,**c**); two distinct angles: (**b**) camera mounted at the back arm of the Barrier Transfer Machine (BTM) and (**c**) camera mounted at the front arm of the BTM for enhanced 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
$\sum_{k,t}$	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
∇l_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 Waihanga New Zealand Infrastructure Commission. New Zealand's Infrastructure Asset Value, Investment, and Depreciation, 1990–2022. 2023. Available online: https://tewaihanga.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.; Hadded, M.; Ghazzai, H.; Touati, H.; Elhadef, 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.; Barmpounakis, 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.

electronics

MDPI

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

$$\mathbf{E}(r,\theta,z,t) = \mathbf{E}(r,\theta)e^{j(\omega t - k_z z)}, \quad (1)$$

$$\mathbf{H}(r,\theta,z,t) = \mathbf{H}(r,\theta)e^{j(\omega t - k_z z)}, \quad (2)$$

check for updates

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

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

$$\begin{pmatrix} \frac{1}{r}\frac{\partial E_z}{\partial \theta} - \frac{\partial E_\theta}{\partial z} \\ \frac{\partial E_r}{\partial z} - \frac{\partial E_z}{\partial r} \\ \frac{1}{r}\frac{\partial (rE_\theta)}{\partial r} - \frac{1}{r}\frac{\partial E_r}{\partial \theta} \end{pmatrix} = -j\omega\mu_0 \begin{pmatrix} \mu H_r - j\kappa H_\theta \\ j\kappa H_r + \mu H_\theta \\ \mu_{rz} H_z \end{pmatrix}, \tag{3}$$

$$\begin{pmatrix} \frac{1}{r}\frac{\partial H_z}{\partial \theta} - \frac{\partial H_\theta}{\partial z} \\ \frac{\partial H_r}{\partial z} - \frac{\partial H_z}{\partial r} \\ \frac{1}{r}\frac{\partial (rH_\theta)}{\partial r} - \frac{1}{r}\frac{\partial H_r}{\partial \theta} \end{pmatrix} = j\omega\varepsilon_0\varepsilon_{rf} \begin{pmatrix} E_r \\ E_\theta \\ E_z \end{pmatrix}. \tag{4}$$

Putting Equations (1) and (2) in Equations (3) and (4), and conducting some manipulation, we obtain

$$E_r = \frac{k_z}{K_c^2}\left(-jK_{c\mu}^2\frac{\partial E_z}{\partial r} + F\frac{1}{r}\frac{\partial E_z}{\partial \theta}\right) - \frac{1}{K_c^2}\left(A_1\frac{\partial H_z}{\partial r} + jA_2\frac{1}{r}\frac{\partial H_z}{\partial \theta}\right), \tag{5}$$

$$E_\theta = -\frac{k_z}{K_c^2}\left(F\frac{\partial E_z}{\partial r} + jK_{c\mu}^2\frac{1}{r}\frac{\partial E_z}{\partial \theta}\right) + \frac{1}{K_c^2}\left(jA_2\frac{\partial H_z}{\partial r} - A_1\frac{1}{r}\frac{\partial H_z}{\partial \theta}\right), \tag{6}$$

$$H_r = \frac{\omega\varepsilon_0\varepsilon_{rf}}{K_c^2}\left(F\frac{\partial E_z}{\partial r} + jK_{c\mu}^2\frac{1}{r}\frac{\partial E_z}{\partial \theta}\right) - \frac{1}{K_c^2}\left(jk_zK_{c\mu}^2\frac{\partial H_z}{\partial r} - Fk_z\frac{1}{r}\frac{\partial H_z}{\partial \theta}\right), \tag{7}$$

$$H_\theta = \frac{\omega\varepsilon_0\varepsilon_{rf}}{K_c^2}\left(-jK_{c\mu}^2\frac{\partial E_z}{\partial r} + F\frac{1}{r}\frac{\partial E_z}{\partial \theta}\right) - \frac{1}{K_c^2}\left(Fk_z\frac{\partial H_z}{\partial r} + jk_zK_{c\mu}^2\frac{1}{r}\frac{\partial H_z}{\partial \theta}\right), \tag{8}$$

where some symbols are introduced for convenience, namely the following:

$$\begin{aligned} K_{c\mu}^2 &= \varepsilon_{rf}k_0^2\mu - k_z^2, \\ F &= \varepsilon_{rf}k_0^2\kappa, \\ K_c^2 &= K_{c\mu}^4 - F^2, \\ A_1 &= \tfrac{Fk_z^2}{\omega\varepsilon_0\varepsilon_{rf}}, \\ A_2 &= \tfrac{K_c^2 + k_z^2K_{c\mu}^2}{\omega\varepsilon_0\varepsilon_{rf}}. \end{aligned} \tag{9}$$

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:

$$\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2} + \frac{\omega\mu_0\mu_{rz}K_c^2}{A_2}\right]H_z = -j\frac{k_zF}{A_2}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2}\right]E_z, \tag{10}$$

$$\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2} + \frac{K_c^2}{K_{c\mu}^2}\right]E_z = \frac{jFk_z}{K_{c\mu}^2\omega\varepsilon_0\varepsilon_{rf}}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2}\right]H_z, \tag{11}$$

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

$$\kappa = 0. \tag{12}$$

In this case, Equations (10) and (11) become

$$\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2}{\partial\theta^2}+\frac{\mu_{rz}}{\mu}\left(\varepsilon_{rf}k_0^2\mu-k_z^2\right)\right]H_z=0, \tag{13}$$

$$\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2}{\partial\theta^2}+\varepsilon_{rf}k_0^2\mu-k_z^2\right]E_z=0, \tag{14}$$

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

$$k_z=0. \tag{15}$$

Putting the propagation constant equal to zero in Equations (10) and (11), we obtain

$$\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2}{\partial\theta^2}+\varepsilon_{rf}\mu_{rz}k_0^2\right]H_z=0, \tag{16}$$

$$\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2}{\partial\theta^2}+\frac{\varepsilon_{rf}}{\mu}\left(\mu^2-\kappa^2\right)k_0^2\right]E_z=0. \tag{17}$$

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]

electronics

MDPI

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

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.

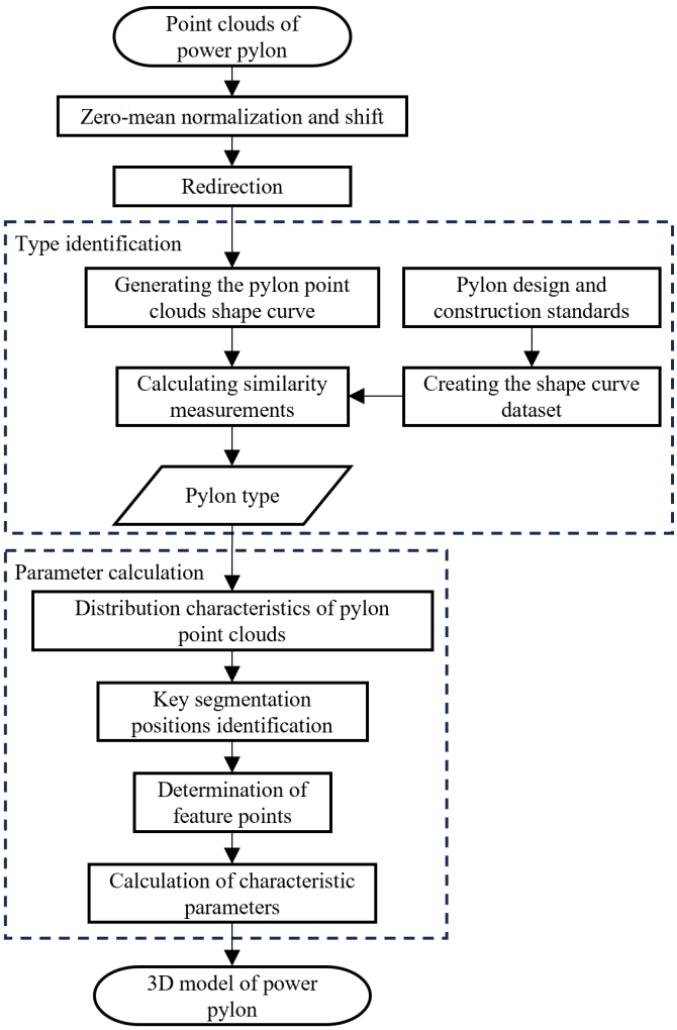


Figure 1. Overall flow chart of type identification and parameter calculation.

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').

$$\begin{cases} x' = x - x_0 \\ y' = y - y_0 \\ z' = z - z_0 - \min(z - z_0) \end{cases} \tag{1}$$

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.

$$\theta = \arccos\left(\frac{v_1}{\sqrt{v_1^2 + v_2^2}}\right) \tag{2}$$

$$\begin{cases} x'' = x'\cos(\theta) - y'\sin(\theta) \\ y'' = x'\sin(\theta) + y'\cos(\theta) \\ z'' = z' \end{cases} \tag{3}$$

where x'', y'', and z'' represent the coordinates of the rotated point clouds.

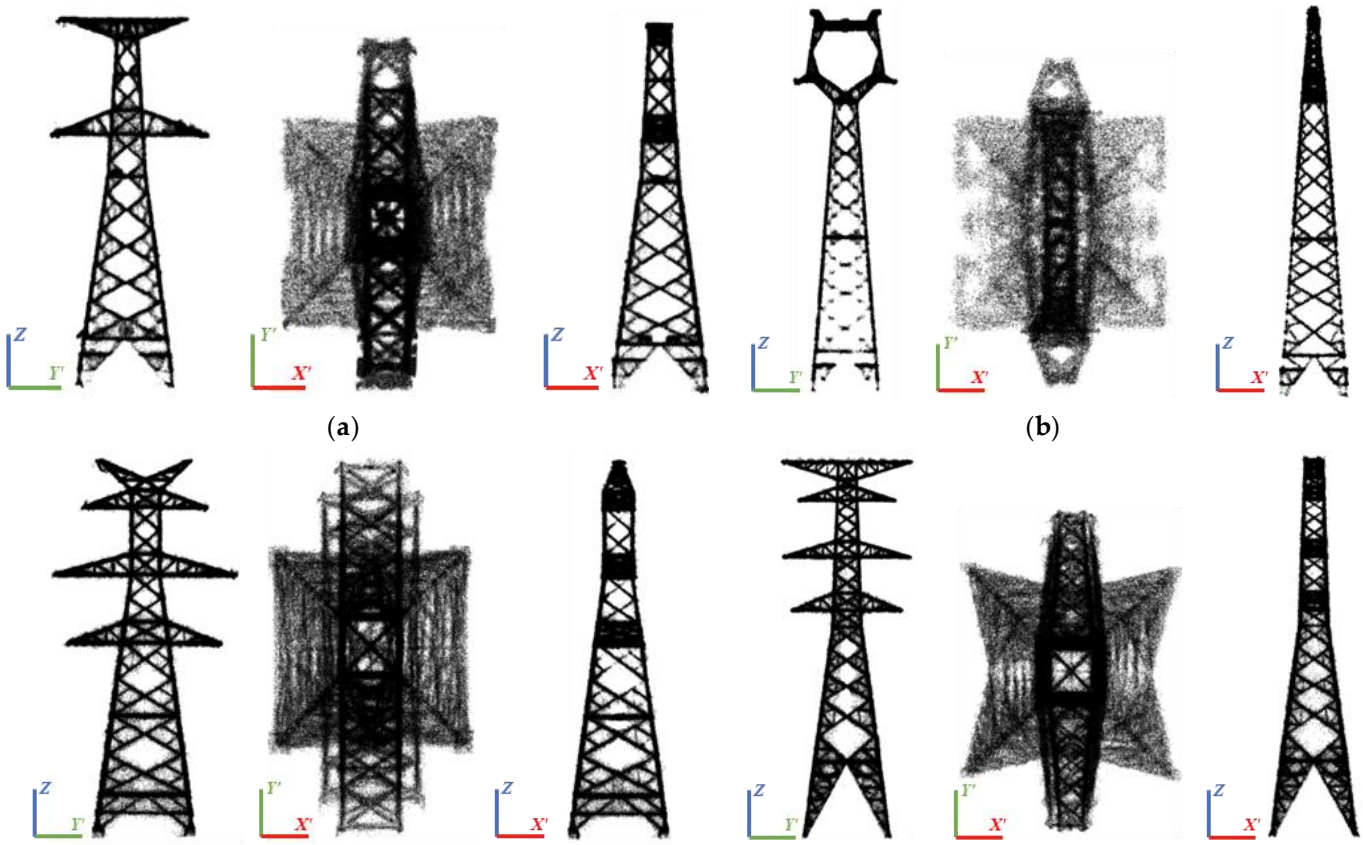


Figure 2. The projections of the redirected pylons on the Y′Z, X′Y′, and X′Z planes. (**a**) Type-a pylon. (**b**) Type-b pylon. (**c**) Type-c pylon. (**d**) Type-d pylon.

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.

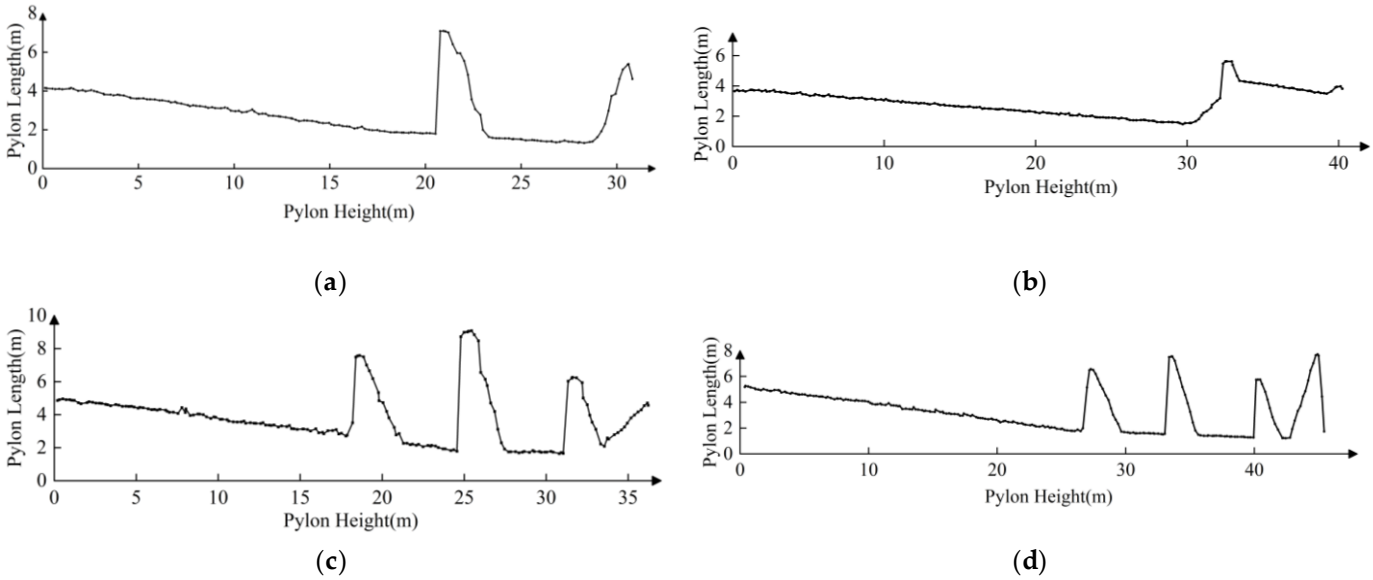


Figure 3. Pylon point cloud shape curve. (**a**) Type-a pylon. (**b**) Type-b pylon. (**c**) Type-c pylon. (**d**) Type-d pylon.

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.

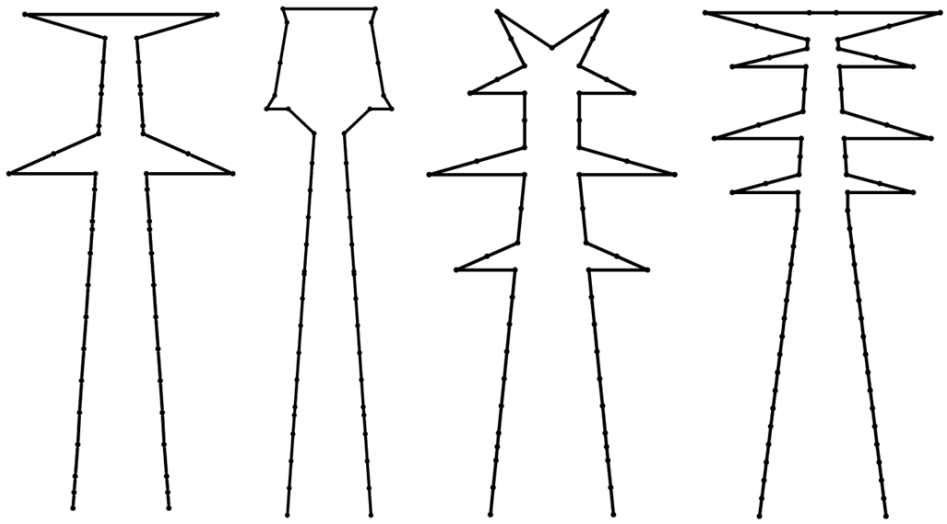


Figure 4. Shape curves of some pylons in the dataset.

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, $X = \{x_1, x_2, \ldots, x_i, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_j, \ldots, y_m\}$. 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.

$$\begin{cases} D(i,j) = d(x_i, y_j) + \min\{D(i-1,j), D(i,j-1), D(i-1,j-1)\} \\ 1 \le i \le n, \quad 1 \le j \le m \end{cases} \tag{4}$$

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.

$$L = \left\{(x_{a_1}, y_{b_1}), (x_{a_2}, y_{b_2}), \cdots, (x_{a_k}, y_{b_k})\right\} \tag{5}$$

Among them, $a_1 = 1$, $b_1 = 1$, $a_k = m$, and $b_k = n$. And to ensure the order of points, for any $i = 1, \ldots, n$, 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.

$$\|L\| = \max_{i=1,\cdots,k} d(x_{a_i}, y_{b_i}) \tag{6}$$

The discrete Fréchet distance between X and Y is defined as follows:

$$D_f(X, Y) = \min\{\|L\|\} \tag{7}$$

$D_f(X, Y)$ 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:

$$\begin{cases} D_h(X, Y) = \max\{d_h(X, Y), d_h(Y, X)\} \\ d_h(X, Y) = \max_{x \in X} \min_{y \in Y} \| x - y \| \\ d_h(Y, X) = \max_{y \in Y} \min_{x \in X} \| y - x \| \end{cases} \tag{8}$$

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.

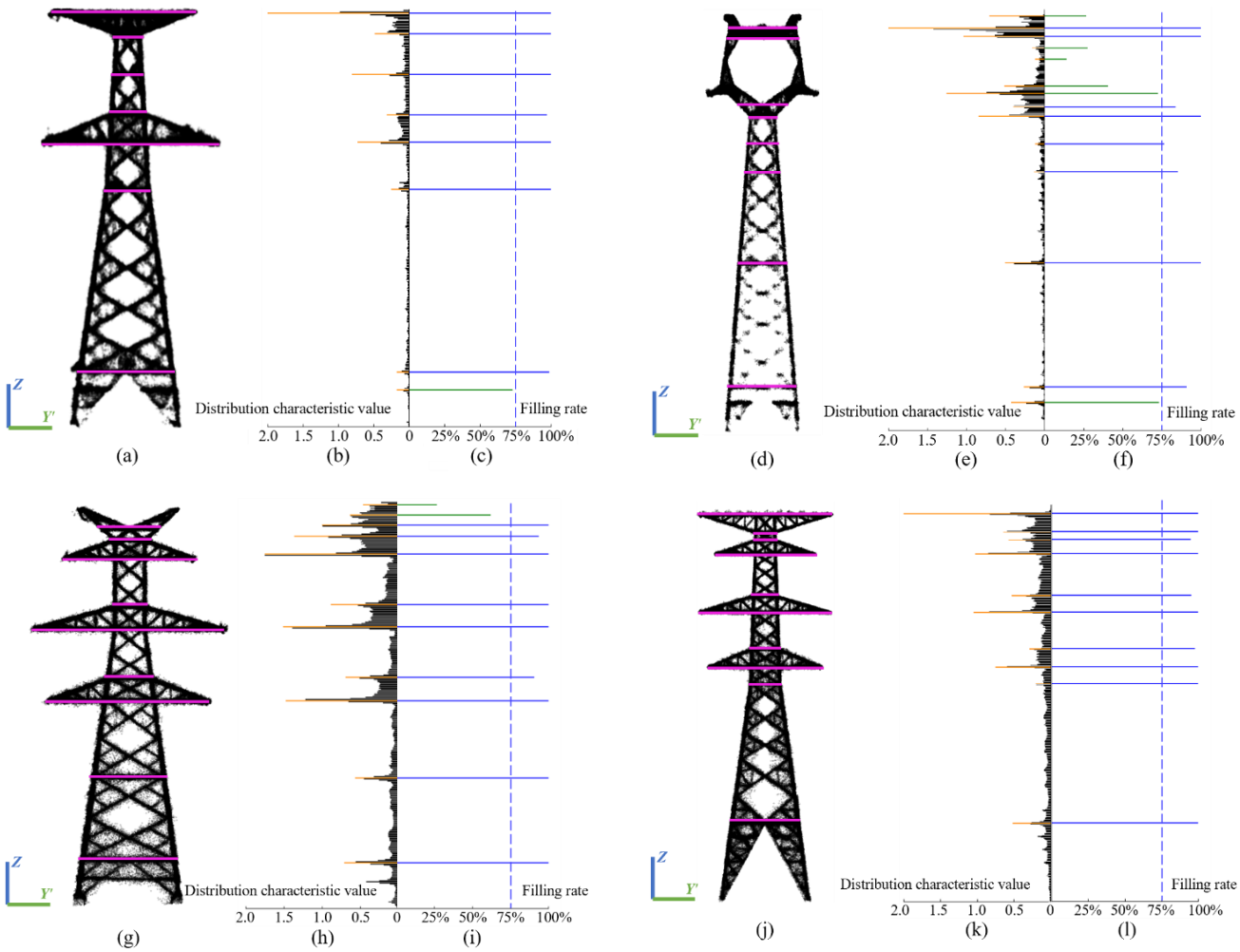


Figure 5. Distribution characteristics of pylon point clouds. (**a**,**d**,**g**,**j**) The projections of the pylon on the Y′Z′ plane. (**b**,**e**,**h**,**k**) Distribution characteristic value histograms, and yellow lines are layers with both the local maximum number of point clouds and the local maximum density. (**c**,**f**,**i**,**l**) Filling rate histograms, and blue lines are the key segmentation positions.

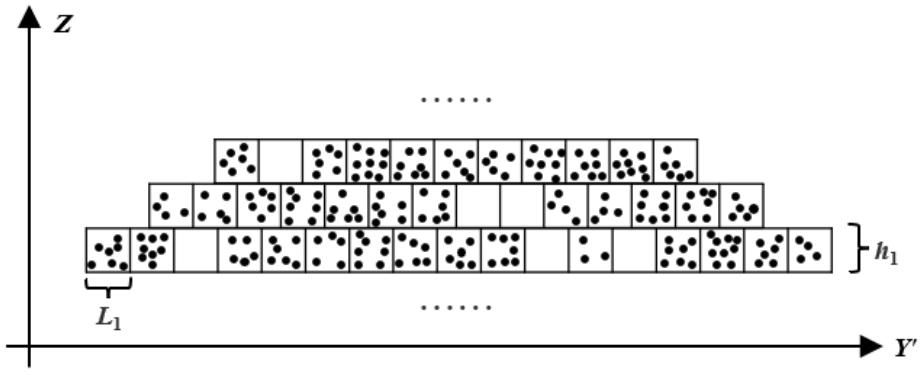


Figure 6. The filling rate calculation process.

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 Ce 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 $G_{i\text{-}1}$ is the segmentation position $S_{i\text{-}1}$ 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.

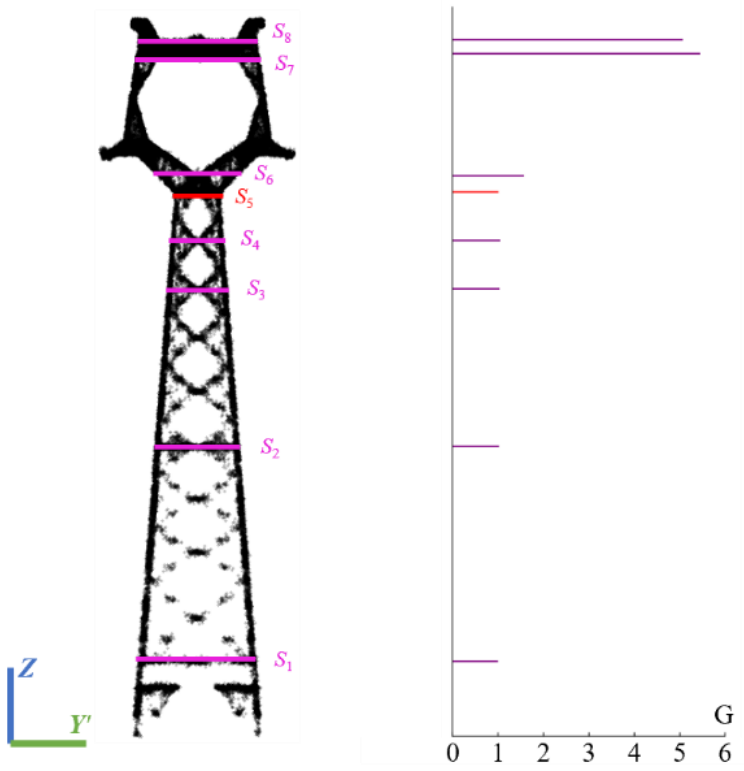


Figure 7. Shape parameters of the type-b pylon.

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.

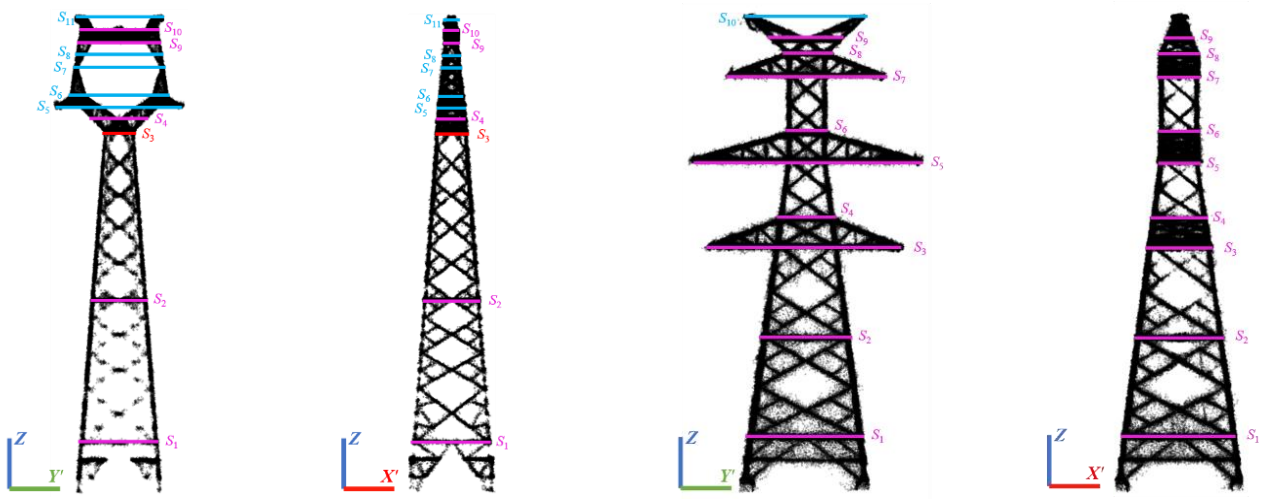


Figure 8. The key segmentation positions of the pylon. (**a**) Type-b pylon. (**b**) Type-c pylon.

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.

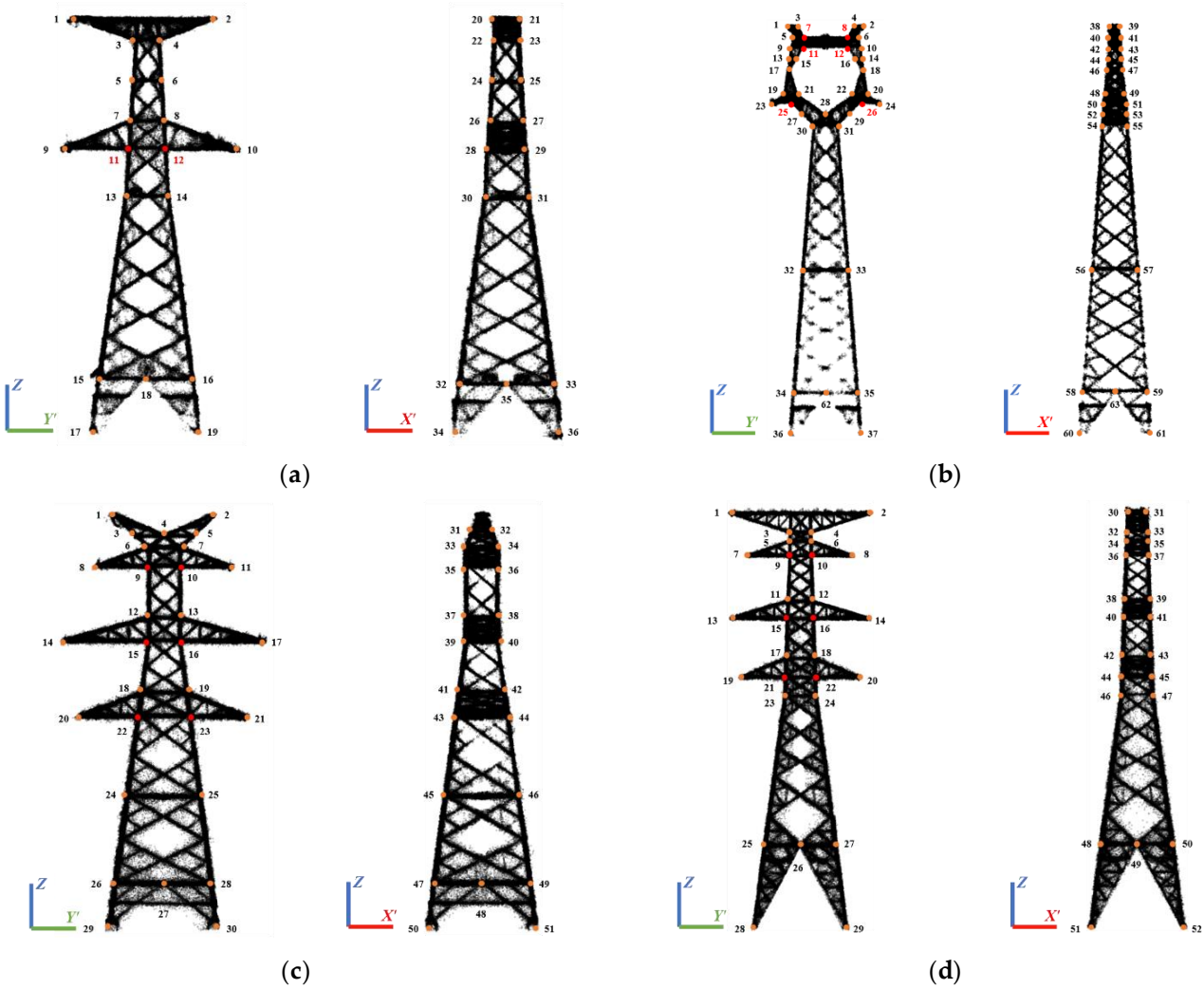


Figure 9. Pylon feature points and numbers, and red points are the feature points obtained by combining with the surrounding orange points. (**a**) Type-a pylon. (**b**) Type-b pylon. (**c**) Type-c pylon. (**d**) Type-d pylon.

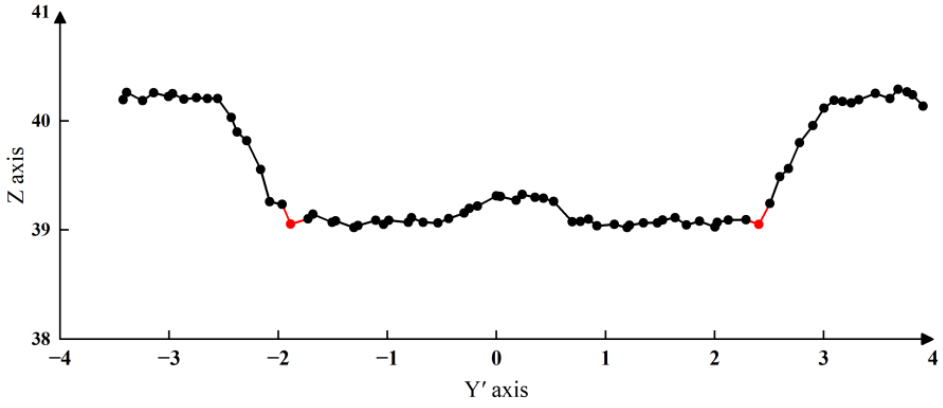


Figure 10. Point cloud distribution curve of the type-b pylon head.

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) × 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%
Ce	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.

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \times 100\% \tag{9}$$

Recall refers to the proportion of pylons of a certain type that are ultimately predicted to belong to that type.

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \times 100\% \tag{10}$$

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.

$$\mathrm{F} = \frac{2\mathrm{P} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}} \times 100\% \tag{11}$$

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.

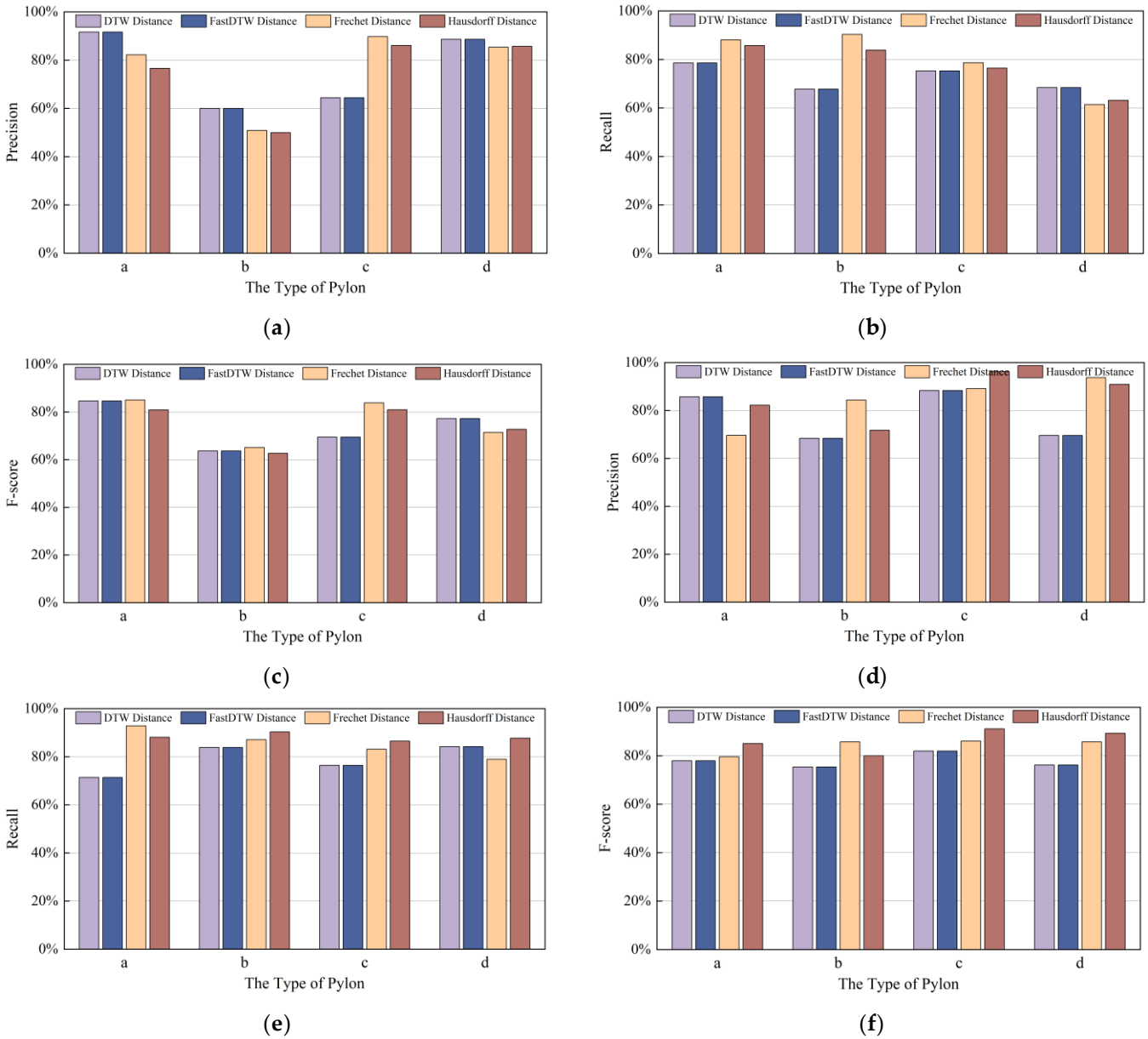


Figure 11. Identification results of pylon types using different similarity measurements. (**a**) The precision of four similarity measurements (original dataset). (**b**) The recall of four similarity measurements (original dataset). (**c**) The F-score of four similarity measurements (original dataset). (**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.

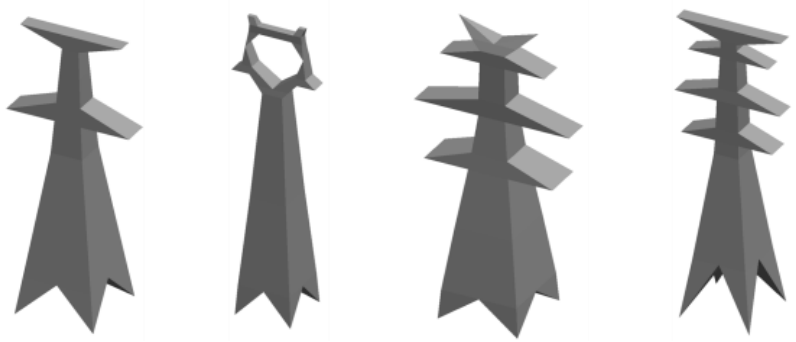


Figure 12. 3D models of pylons.

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.

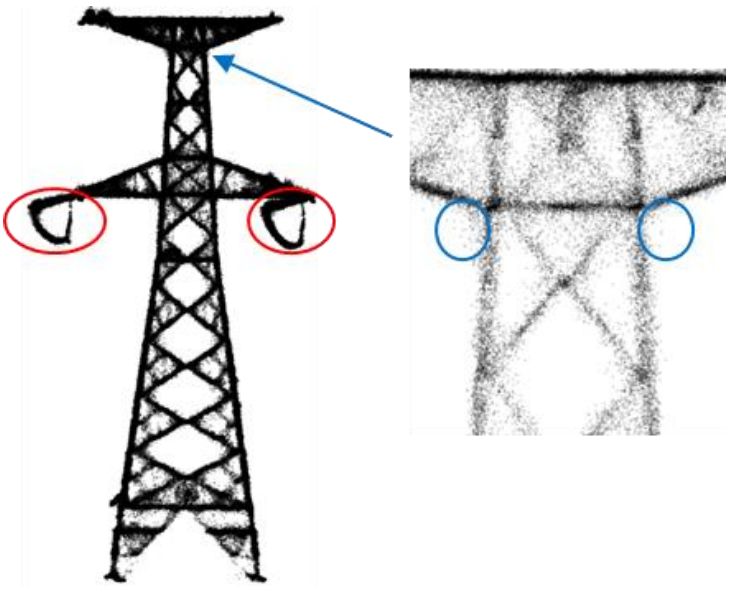


Figure 13. Pylon point clouds with interference points.

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]