electronics

MDPI

Article

Predicting the Characteristics of High-Speed Serial Links Based on a Deep Neural Network (DNN)—Transformer Cascaded Model

Liyin Wu [1], Jingyang Zhou [1], Haining Jiang [1], Xi Yang [1], Yongzheng Zhan [2] and Yinhang Zhang [1,*]

[1] School of Communication and Electronic Engineering, Jishou University, Jishou 416000, China; 2022700814@stu.jsu.edu.cn (L.W.); 2022700812@stu.jsu.edu.cn (J.Z.); 2023700812@stu.jsu.edu.cn (H.J.); ynkej@163.com (X.Y.)

[2] Shandong Yunhai Guochuang Cloud Computing Equipment Industry Innovation Co., Ltd., Jinan 250101, China; sdzyz1989@163.com

* Correspondence: zyh149032@jsu.edu.cn

Abstract: The design level of channel physical characteristics has a crucial influence on the transmission quality of high-speed serial links. However, channel design requires a complex simulation and verification process. In this paper, a cascade neural network model constructed of a Deep Neural Network (DNN) and a Transformer is proposed. This model takes physical features as inputs and imports a Single-Bit Response (SBR) as a connection, which is enhanced through predicting frequency characteristics and equalizer parameters. At the same time, signal integrity (SI) analysis and link optimization are achieved by predicting eye diagrams and channel operating margins (COMs). Additionally, Bayesian optimization based on the Gaussian process (GP) is employed for hyperparameter optimization (HPO). The results show that the DNN–Transformer cascaded model achieves high-precision predictions of multiple metrics in performance prediction and optimization, and the maximum relative error of the test-set results is less than 2% under the equalizer architecture of a 3-taps TX FFE, an RX CTLE with dual DC gain, and a 12-taps RX DFE, which is more powerful than other deep learning models in terms of prediction ability.

Keywords: high-speed link; signal integrity; eye diagram; channel operating margin; cascaded model

Citation: Wu, L.; Zhou, J.; Jiang, H.; Yang, X.; Zhan, Y.; Zhang, Y. Predicting the Characteristics of High-Speed Serial Links Based on a Deep Neural Network (DNN)—Transformer Cascaded Model. *Electronics* **2024**, *13*, 3064. https://doi.org/10.3390/electronics13153064

Academic Editor: Shing-Hong Liu

Received: 1 July 2024
Revised: 24 July 2024
Accepted: 31 July 2024
Published: 2 August 2024

1. Introduction

As the transmission bandwidth of wireline serial link technology reaches the GHz level, it is no longer possible to ensure efficient signal transmission by simply optimizing the dielectric and layout structure. High-speed serial link systems suffer from serious signal integrity (SI) problems due to the skin effect, dielectric loss, crosstalk, reflections, and jitter; therefore, the SI analysis becomes more and more strict in the design stage of high-speed serial links. The simulation analysis of SI usually consists of two steps: electromagnetic field solvers (EMFSs) and circuit system simulation [1]. Firstly, an EMFS is used to obtain S-parameters to characterize the frequency response of the circuit. Then, these S-parameters are imported into the model circuit system for time-domain simulation to obtain the main SI metrics, including an eye diagram, the impulse response, and transient waveforms.

Although the traditional SI analysis based on physical models of high-speed links can offer high accuracy, it consumes a lot of time and computer resources. Input/Output Buffer Information Specification Algorithmic Model Interface (IBIS-AMI) is a behavioral model that simulates the input/output behavior and algorithms of end-to-end high-speed serial links, simplifying the internal physical details [2]. Compared to physical models, it has the advantages of speed, simplicity, and low resource consumption, but it has poor accuracy and lacks flexibility. Compliance standard template comparison is another commonly used method [3]. This can swiftly and intuitively estimate channel performance, but it inevitably

discards the channel margins due to the need to strictly satisfy discrete metrics [4]. On the contrary, the channel operating margin (COM) technique can effectively overcome this disadvantage by searching the optimal design space of the whole link in the form of a signal-to-noise ratio (SNR). Compared to the traditional SI metrics such as eye diagrams and the bit error rate (BER), the COM approach shows advantages such as simpler operation, faster speed, and more efficient testing. Obviously, compared with Annex69B channel evaluation for 10 Gb/s Ethernet (10 GbE), using COMs is more accurate [5]. Although the COM approach can provide an effective way to evaluate and optimize high-speed links, it requires numerous iterations for spatial search and is not flexible enough to analyze high-capacity channels.

Machine learning (ML) has been widely used in the simulation and design of high-speed links in recent years, and shows the ability to improve the efficiency of SI analysis. The authors of [4,6–10] searched for features from simulation data and trained Artificial Neural Network (ANN), Deep Neural Network (DNN), and Least-Squares Support Vector Machine (LS-SVM) models to replace circuit system simulations for the accurate prediction of time-domain (TD) and frequency-domain metrics such as eye height (EH)/eye width (EW) and return loss (RL)/insertion loss (IL); however, the acquisition of simulation data consumes a lot of computational resources and time. Feedforward Neural Networks (FNNs), Random Forest Regression (RFR), and Support Vector Machines (SVMs) are used to achieve simple predictions of simulation data [11–13], such as for predicting S-parameters and impulse responses. When dealing with more complex simulation data, traditional ML methods may lose many features. The Recurrent Neural Network (RNN) is an ML model that can effectively capture the features of complex data, especially for sequence data such as S-parameters and impulse responses. The authors of [14–20] employ RNN and Long Short-Term Memory (LSTM) architecture to create surrogate models for predicting the transient response waveforms of complex high-speed links. These independent surrogate models can perform SI analysis based on simulation data, but cannot deal with the physical parameters of links. SI analysis based on physical parameters requires more complicated ML methods, and the authors of [21–24] achieved effective predictions of physical parameters used for assessing high-speed link performance by combining multiple deep learning algorithms. Moreover, balanced architecture optimization for high-speed serial links can be achieved based on ML [4,25–27]. In conclusion, performance prediction and architecture optimization [28] are two important parts of ML applications for high-speed serial links.

In this paper, a DNN–Transformer cascaded model is proposed for the SI analysis and optimization of high-speed serial links. This model can skip the use of EMFSs and circuit system simulation, and directly predicts SI metrics including EH/EW, IL/RL, impulse response, and COM values according to physical parameters. Meanwhile, the optimization of links can be achieved by using this model to predict the corresponding COM values and equalizer parameters for the links with different equalizer configurations. Furthermore, Bayesian optimization based on the Gaussian process (GP) is used to optimize the hyperparameters under the same conditions for different combinations of models. Compared with the prediction of high-speed link performance using traditional ML [6–10], DNN–Transformer can directly use the physical parameters of the link for analysis, and its prediction accuracy is significantly better. For the prediction of simulation data such as impulse responses, DNN–Transformer can capture more features and its prediction ability is more accurate than that achieved when using an RNN [14] or LSTM [20] model alone to create a surrogate model. The results show that the DNN–Transformer model can achieve more effective prediction. The performance and feasibility of the proposed method is illustrated by the prediction data and graphical results given in this paper.

The main contributions of this paper can be summarized as follows:

(1) Based on the key physical parameters of channels, neural network models are used to directly analyze link performance, and this obviates the time-consuming processes of EMF solving and circuit system simulation.

(2) Through neural network models, the accurate prediction of multiple SI indicators and equalizer parameters is achieved. Additionally, we show that SI analysis and link optimization can also be rapidly achieved.
(3) A DNN–Transformer cascaded model is proposed, Bayesian optimization is used to tune the hyperparameters of this model, and its superior performance is demonstrated by comparing it with other models.

The rest of this paper is organized as follows: Section 2 discusses the basic principles of this work, including the principles of the SI and COM methods. Section 3 describes how the simulation dataset was created. In Section 4, the design idea of the DNN–Transformer model and test metrics are presented. Section 5 gives the numerical results to demonstrate the prediction performance of our model. Finally, Section 6 concludes the paper.

2. Fundamental Methods

2.1. Signal Integrity and S-Parameters

As transmission rates increase, the inefficiencies of traditional parallel signal transmission methods are becoming apparent and high-speed serial digital transmission is becoming widely used. Figure 1 shows the widely used SerDes structure in a high-speed serial link system. It comprises a transmitter chip with a serializer and a feedforward equalizer (FFE), a channel, and a receiver chip with a decision feedback equalizer (DFE) and a de-serializer. The FFE at the sending end mainly performs pre-emphasis on the signal to improve its high-frequency component. The composition of a wireline channel is complex, usually including traces, vias and connectors, etc. These physical structures exhibit low-pass filter characteristics and can enlarge the high-frequency loss of a signal, severely worsening the signal transmission quality and leading to SI problems. The continuous-time linear equalizer (CTLE) at the receiving end is mainly used to compensate for the high-frequency loss of the channel, eliminate the pre-cursors, and suppress the trailing of the pulse response. The DFE is mainly used to eliminate the post-cursors of the pulse response. The combination and parameter settings of the FFE, CTLE, and DFE have a significant impact on the SI of ultra-high-speed wireline serial communication links.

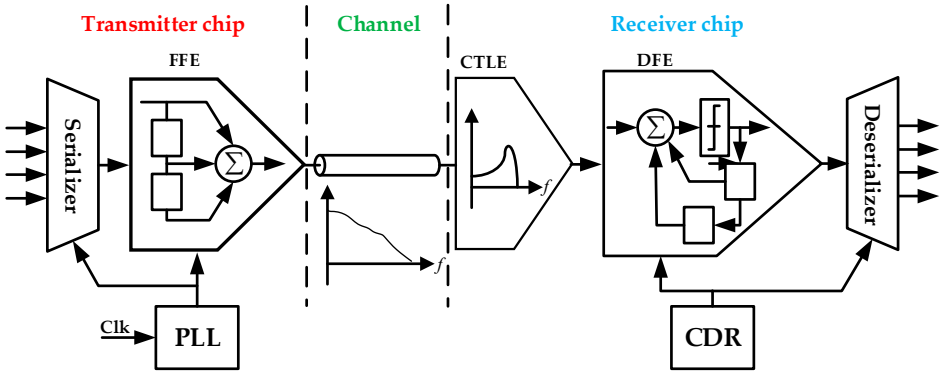


Figure 1. Common structure of a SerDes circuit.

The main goal of SI analysis is to detect and reduce the factors that cause losses, such as dithering, reflection, and crosstalk. On the one hand, the SI can be analyzed from the frequency domain, as shown in Figure 2a. This mainly includes IL, RL, insertion loss deviation (ILD), and the insertion loss-to-crosstalk ratio (ICR), which are subsequently placed into a template for evaluation. On the other hand, the SI can be evaluated in the time domain, such as through eye diagrams, transient simulations, bathtub curves, and bit error rates, as shown in Figure 2b.

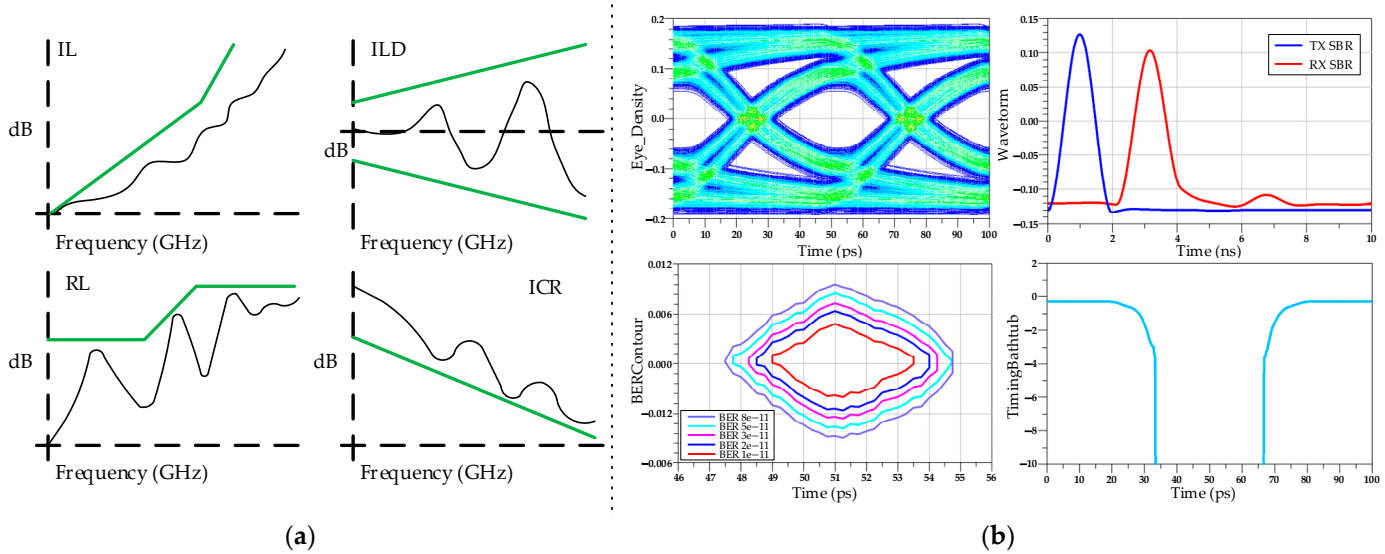


Figure 2. (**a**) FD. (**b**) TD. Illustration of two different simulation methods for detecting SI.

The critical method of traditional SI analysis involves the acquisition of S-parameters. These parameters contain comprehensive Frequency Domain (FD) characteristics of the transmission channel and offer a large amount of information on aspects such as reflection, crosstalk, and loss. Moreover, S-parameters can be employed in time-domain simulations to generate data such as eye diagrams and bathtub curves.

2.2. Channel Operating Margin

The COM method is a high-speed serial link characterization method recommended by the IEEE 802.3 working group for channel compliance testing. The official definition states that a "COM is a figure of merit (FOM) for channels determined from a minimum reference PHY architecture and channel s-parameters" [29]. This approach provides a relatively accurate and fair environment for the physical design of channels by considering various factors such as loss, reflection, inter-symbol interference (ISI), dispersion ISI, crosstalk, and device specifications, enabling a relatively accurate assessment of channel performance and the impact of aggressor channels on victim channels. The impact of equalizers has been considered in subsequent versions of the COM method, and the value of a COM based on the FOM can be improved by selecting equalizer parameter settings. Consequently, calculating the COM can also determine whether the channel quality meets the transceiver's SI requirements [30]. As shown in Equation (1), the COM can be expressed by the ratio of available signal amplitude A_s to statistical noise amplitude A_n.

$$\mathrm{COM} = 20 \times \log_{10}\left(\frac{A_s}{A_n}\right) \tag{1}$$

The process of deriving the COM involves several essential steps, including determining the transfer function, converting the transfer function into the impulse response, applying transmitter and receiver equalizer algorithms, and performing statistical noise calculations. Figure 3 presents a flowchart of COM model derivation and depicts two paths for the victim and aggressor channels. Additionally, the model also considers transmitter and receiver package interference S (tp/rp), termination resistance Rd, equalizers, and filters. To better simulate actual channels, the COM model incorporates Gaussian white noise and jitter at the receiver end. Figure 4 shows the detailed process of COM value calculation.

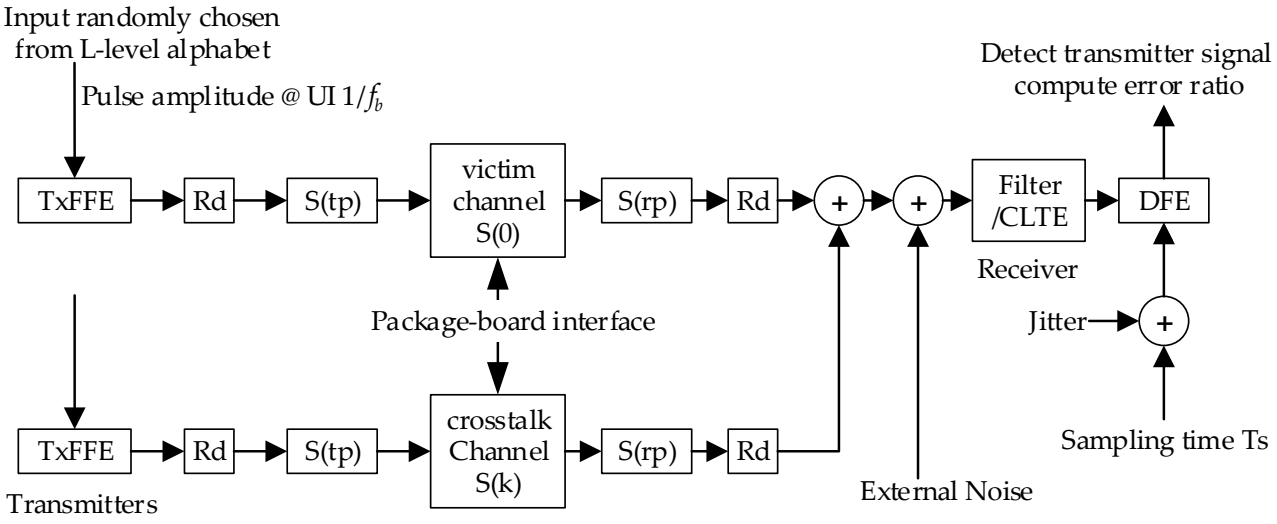


Figure 3. A typical COM model.

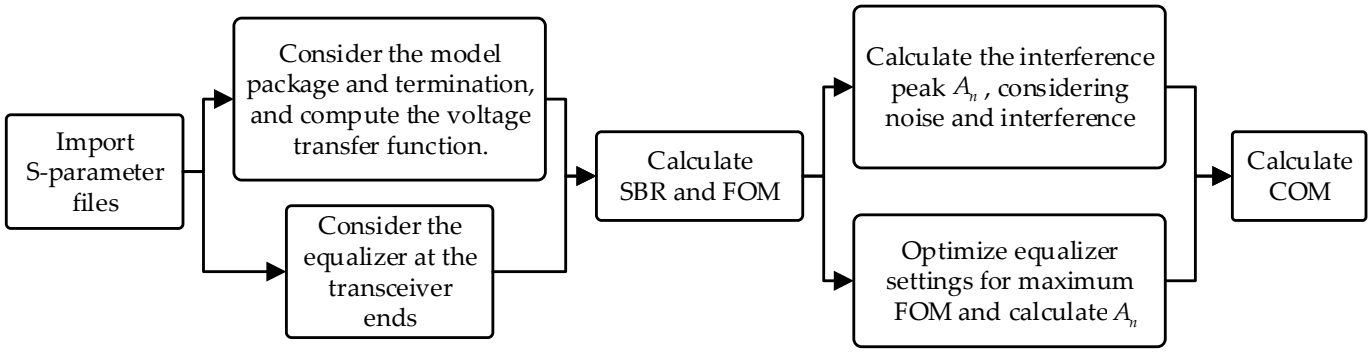


Figure 4. COM calculation process.

The equalizer parameters are mainly set based on the channel loss. The FOM serves as a quantitative metric for assessing channel quality and equalizer performance. It takes into account several factors affecting SI, including ISI, jitter, crosstalk, and noise. The FOM calculation formula can be expressed as follows:

$$\mathrm{FOM} = 10 \times \log_{10}\left(\frac{A_s^2}{\sigma_{TX}^2 + \sigma_{ISI}^2 + \sigma_J^2 + \sigma_{XTK}^2 + \sigma_N^2}\right) \tag{2}$$

The numerator A_s is derived from the amplitude of the impulse response at t_s, which corresponds to the main cursor of the impulse response $h(t)$. The denominator includes the sum of variances from all noise, jitter, and interference components. σ_{TX}^2 represents the noise variance at the transmitter, σ_{ISI}^2 denotes the variance in the residual ISI amplitude, and σ_J^2 indicates the variance in jitter amplitude. In COM analysis, jitter is accounted for by converting horizontal jitter into vertical noise at the sampling instant t_s. Additionally, σ_{XTK}^2 represents the total crosstalk variance from all interference paths, while σ_N^2 denotes the Gaussian white noise at the receiver sampling point.

The COM approach entails exploring all possible parameter combinations of the TX and RX equalizers within a set range to find the configuration that maximizes the FOM. This process determines the values of A_s and A_n that result in the optimal FOM.

Traditional methods rely on various indicators such as jitter, eye height, and eye width, but the COM serves as a comprehensive metric for evaluating a serial link. It can significantly reduce the computation time and number of iterations, and provides an accurate evaluation of channel performance both before and after equalization.

3. Dataset Construction

3.1. Channel Design and Dataset Splitting

In this work, the channel datasets were generated using professional PCB design software. As shown in Figure 5, Altium Designer 20.0.13 and Allegro 17.4 (EDA software) were primarily utilized to create the layouts for high-speed differential lines. Subsequently, the PCB layout files were imported into Advanced Design System 2021 (ADS) or HFSS 2023 R1 for electromagnetic (EM) simulation. Our design conforms to the specifications for USB end-to-end differential transmission lines, including parameters such as line width and length, the characteristic impedance of transmission channels, and differential line spacing.

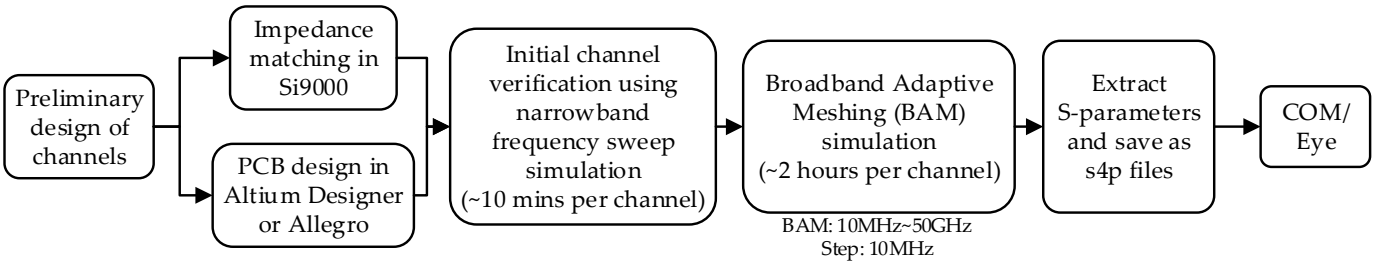


Figure 5. Channel fabrication and simulation process.

The differentiated characteristics of channel data, as presented in Table 1, primarily include parameters such as material, trace dimensions, and PCB type. The board characteristics encompass both numerical features—such as permittivity (*Er*), the dissipation factor (*Df*), and board thickness—and categorical features such as the PCB type. *Er* consists of the sheet permittivity's real part and the dielectric loss tangent angle (*TanD*). *Df* is derived from the Djordjevic model, which is defined by parameters including the low and high frequencies of *TanD* and relative high-frequency permittivity. In this study, *Df* is quantified according to the numerical values of *TanD* and its corresponding high and low frequencies. The PCB type is categorized into either the stripline type, represented by a numerical value of 1, or the microstrip type, represented by 0.

Table 1. Channel feature parameters and quantitative methods.

Feature Variable	Input	Output
Er	PCB sheet permittivity real part and TanD	Eye diagrams COM values Equalizer parameters
Df	PCB sheet dissipation factor	
Trace	PCB trace length	
	PCB trace width	
	Differential line space	
	PCB trace thickness	
	PCB trace conductivity	
	PCB trace structure (profile)	
Zd	Differential impedance	
PCB Type	Stripline is 1 and microstrip is 0	

These trace features are quantified based on specific parameters such as the trace length, width, thickness, conductivity, profile, and differential line spacing. The profile is characterized by the geometric shape of the trace, such as the number of corners. The other characteristics are directly represented by their corresponding numerical values.

3.2. COM Configuration and Equalizer Setting

This work utilizes the COM 4.0 version program provided by IEEE 802.3. The configuration references the enhanced COM (eCOM) simulation setup with the USB4 Gen4 standard and incorporates the elements of the 50 GBASE-KR standard. The PAM4 configuration settings, as shown in Table 2, employ a combination of a transmitter FFE and receiver-side CTLE and DFE. This configuration includes the tap coefficients of the TX FFE and RX DFE, the adaptive range of DC gain, and the zero-pole positions of the RX CTLE. Additional parameters include the number of signal levels, denoted by L. For PAM4 signals, L is set to 4. According to the recommended PAM4 setup based on USB4, the symbol rate is set to 20 GBd, and the differential peak voltage output of the transmitter is set to 0.4 V, denoted as A_v. The USB4 Gen4 standard suggests that PAM3 should be used as the modulation method. The COM settings for PAM3 are roughly same as those of PAM4, except that the symbol rate is set to 25.6 GBd, and L is set to 3.

Table 2. Configuration of COM parameters.

Parameter	Symbol	Setting	Unit
Symbol rate	f_b	20	GBd
Number of signal levels	L	4	
Samples per UI	M	32	
Target detector error ratio	DER_0	10^{-8}	
Transmitter output voltage, victim	A_v	0.4	V
CTLE DC gain	g_{DC}	[−12:1:0]	dB
CTLE DC gain2	g_{DC_HP}	[−6:1:0]	dB
CTLE HP pole	f_{HP_PZ}	0.25	GHz
CTLE zero	f_Z	$f_b/2.5$	GHz
CTLE pole1	f_{P1}	$f_b/2.5$	GHz
CTLE pole2	f_{P2}	f_b	
FFE main cursor	c (0)	0.62	
FFE pre-cursor	c (−1)	[−0.18:0.02:0]	
FFE post-cursor	c (1)	[−0.38:0.02:0]	
DFE length	N_b	12	UI
DFE magnitude limit	$b_{\max}(1)$	0.75	
	$b_{\max}(2 \sim N_b)$	0.2	
COM pass threshold	th	3	dB

Based on the adopted configuration of TX FFE + RX CTLE + RX DFE, the prediction parameters include tap coefficients of the FFE, DC gain of the CTLE, and tap coefficients of the DFE. The TX FFE and RX CTLE data are obtained by searching for FOM values throughout the design space, while the DFE coefficients are determined according to the pulse response $h(t)$ corresponding to the optimal FOM. The SI is significantly impacted by post-cursors closer to the main cursor, while the more distant post-cursors exert minimal influence, so only the first four post-cursors of the DFE are predicted.

3.3. Dataset Splitting

For this work, a total of 325 channels were collected. In this dataset, 280 channels were used to create an eye diagram dataset, referred to as Dataset A, and 215 channels were used to create a dataset for the COMs and predicate the parameters of the equalizers, referred to as Dataset B. Due to the impact of the FOM, the COM approach uses different equalizer parameter optimization methods for stripline and microstrip systems, which results in

a weakened fitting ability of the ML model. Consequently, we standardized Dataset B using strip-line channels. Among the 215 channels in Dataset B, 170 channels are stripline channels in Dataset A, while 45 channels are newly created stripline channels. According to standard methodology, both datasets are partitioned into training, validation, and test sets. Part of the test set in Dataset B is extracted from the validation set. The split settings are detailed in Table 3.

Table 3. Dataset split settings.

	Number of Channels	
Datasets	A: 280	B: 215 (170 from A)
Training Set	184	140
Validation Set	40	40
Testing Set	56	40 (5 from the validation set)
Total	325	

4. Construction of the Cascaded Model and Training

4.1. DNN

The M-P model, proposed by psychologist McCulloch and logician Pitts, is a mathematical model that was developed through an analysis and synthesis of the basic properties of neurons [31]. This model is crucial for the implementation of neural networks and constitutes the foundational unit of a DNN. The structure of this model is depicted in Figure 6. The $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T$ represents the n input features of the neuron, and $\boldsymbol{\omega}_i = (\omega_{i1}, \omega_{i2}, \cdots, \omega_{in})$ denotes the weight vector responsible for linear weighted connections. The weight vector $\boldsymbol{\omega}$ is optimized within neural networks to enhance the accuracy of predictions. The linear weighted output $\boldsymbol{Z}$ can be expressed as follows:

$$Z_i = \sum_{j=1}^{n} \omega_{ij} x_j \tag{3}$$

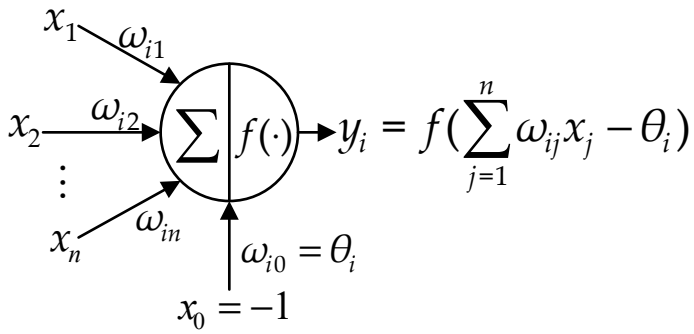


Figure 6. The structure of the M-P neuron model.

Figure 6 depicts the structure of the M-P neuron model. The difference between Z and an offset θ is used as the input to the function $f(\cdot)$. The function $f(\cdot)$ is referred to as the activation function and used to derive the output of the neuron. θ can be considered as the weight of input x_0 with a fixed value of -1 and is also an optimization target of the neural network. In this paper, the ReLU function is adopted as the activation function because it can more effectively mitigate the occurrence of overfitting. The formula for this is shown in Equation (4).

$$f_{\mathrm{ReLU}}(x) = \max(0, x) \tag{4}$$

The output of the activation function is typically used as the input for adjacent layers. DNNs can complete tasks such as regression, classification, and recognition. DNNs are widely used for regression tasks in production and performance evaluation [4,6–9,32]. The

DNN used in this paper, as shown in Figure 7, adopts a standard structure with multiple hidden layers, along with an input layer and a linear regression output layer [33], and employs a fully connected network architecture between its neurons.

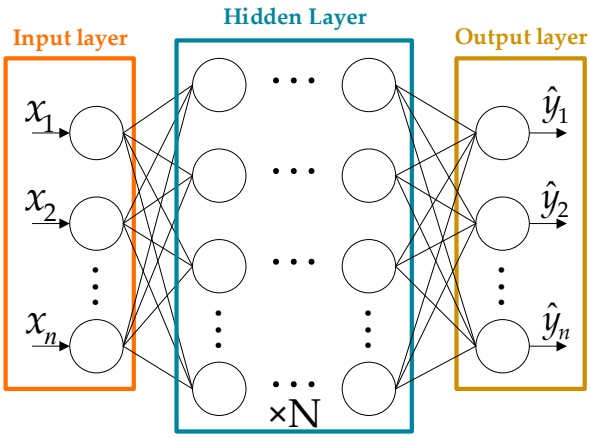


Figure 7. The structure of this study's DNN.

In order to make the model converge better, the training model usually requires a cost function to evaluate the gap between model-predicted values and actual values. In this paper, Smooth L1 [34] is adopted as the cost function:

$$\text{smooth}_{\text{L1}} = \begin{cases} 0.5 \times (y_n - \hat{y}_n)^2 & \text{if} |y_n - \hat{y}_n| < 1 \\ |y_n - \hat{y}_n| - 0.5 & \text{otherwise} \end{cases}, \tag{5}$$

where y_n and $\hat{y}_n$ represent the real value and predicted value, respectively.

The performance of the model is evaluated using the cost function; then, an optimizer is employed to backpropagate and continuously minimize the cost function. The model parameters and weights are continuously updated until the expected performance is achieved. After comparing several different optimizers, the Adam optimizer was ultimately selected for use in this paper [35].

4.2. Transformer for Regression

The accuracy of equalizer parameters and COM values directly predicted by channel feature parameters is relatively low. In this work, the Single-Bit Response (SBR) before and after equalization is first predicted based on the feature parameters of the channel, and then the equalizer parameters and COM values are determined according to the predicted pulse response. The pulse response needs to be used as a time series for prediction. Traditionally, RNN or LSTM models are used for time-series prediction, but these networks have certain limitations. The Transformer model can effectively improve the gradient explosion problem that occurs in LSTM by using attention mechanisms, thereby enhancing prediction precision [36].

The formula of an attention mechanism is shown in Equation (6). It mainly consists of three inputs: Q (Query), K (Key), and V (Value), along with a softmax module. When Q, K, and V are identical, it is referred to as a self-attention mechanism.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{6}$$

The complete Transformer architecture primarily includes an embedding layer, a positional encoding layer, an encoder layer, a decoder layer, and an output layer. In this work, since our task is a time-series prediction task rather than an NLP task, the positional encoding layer has been removed. The essence of time-series prediction can be viewed as a regression task; the model only needs to generate a single output value corresponding

to each time point and does not need a decoder to generate sequence outputs. Using only the encoder layer is sufficient, which simplifies the model structure and reduces the computational load. As shown in Figure 8, our final design keeps the embedding layer, the encoder layer, and the final output layer. Additionally, in order to achieve faster convergence, the Gaussian Error Linear Unit (GELU) was chosen as the activation function.

$$f_{\mathrm{GELU}}(x) = xP(X \leq x) = x \times \frac{1}{2}[1 + \mathrm{erf}(x/\sqrt{2})] \tag{7}$$

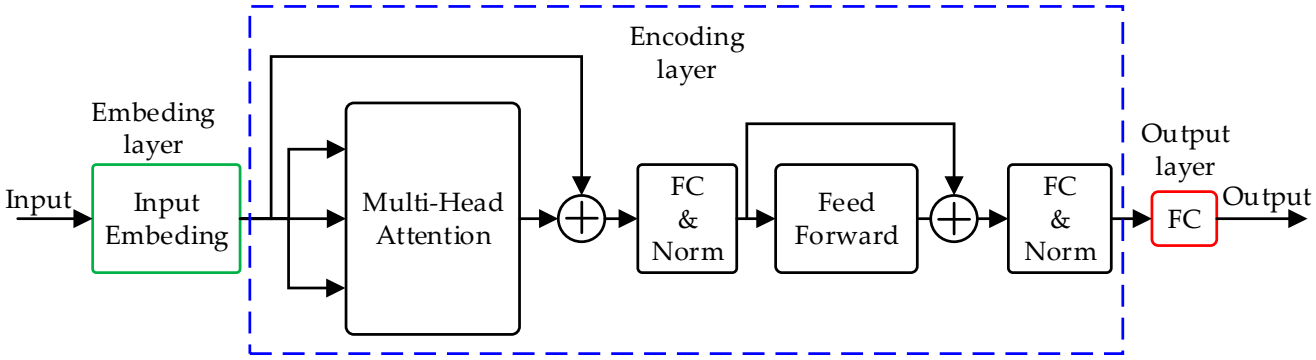


Figure 8. The structure of our Transformer for regression tasks.

4.3. Cascaded Model and Training

As illustrated in Figure 9, a cascaded Transformer and DNN model is utilized in this work to accurately predict the COM value and its corresponding equalizer coefficients. A total of 325 channel features have been collected as a dataset, and a DNN model is used to predict the spectral feature for each channel, specifically the IL and RL at the Nyquist frequency points. These predicted spectral data, along with the channel features, are then fed into a Transformer model to derive the pre-equalization SBR. Utilizing the pre-equalization SBR, the corresponding equalizer parameters under the COM configuration are calculated by the DNN model. In order to compute the post-equalization SBR, equalizer parameters combined with the channel features and frequency-domain features are then sent into another Transformer model. Finally, the sampled post-equalization SBR, channel features, and spectral features are used to compute the final corresponding COM value in the DNN model.

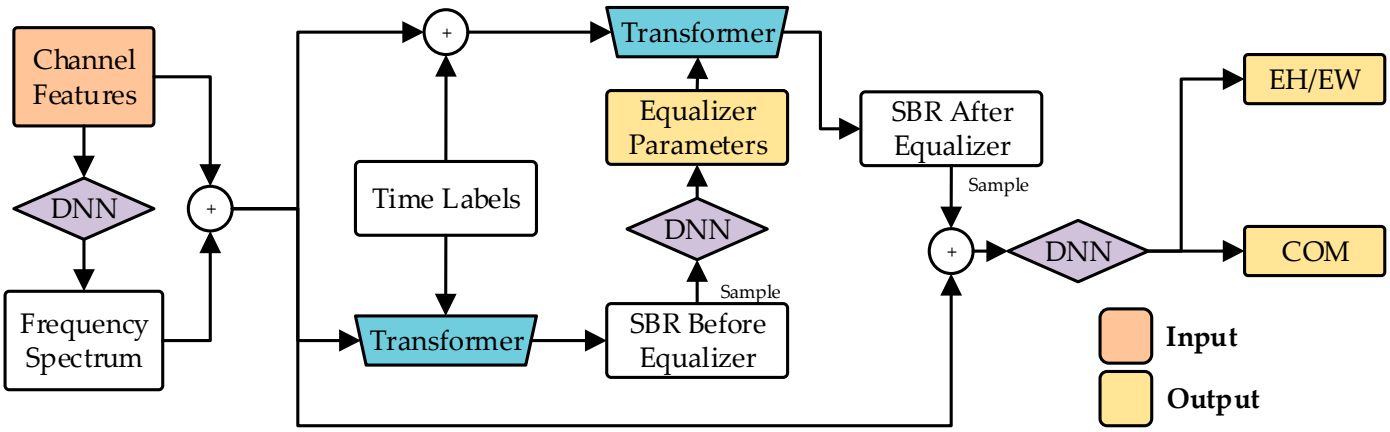


Figure 9. The structure of the cascaded model.

The input vector of physical characteristics for each channel can be defined as follows:

$$\boldsymbol{X} = \{x_{ij}\}_{i=1:N_C, j=1:N_P}, \tag{8}$$

where N_C represents the number of channels in the dataset, N_P is the total number of channel features, and $\boldsymbol{X}$ is an $N_C \times N_P$ matrix. The element x_{ij} denotes the quantified value of a specific physical feature of a channel: for instance, x_{11} corresponds to the length feature value of first channel.

The DNN used in the cascaded model can be represented by Equation (9):

$$\begin{aligned} \boldsymbol{Y} &= F_{DNN}(\boldsymbol{Z}, \boldsymbol{\theta}) \\ &= \boldsymbol{W}^{(l)} f_{\mathrm{ReLU}}(\cdots(\boldsymbol{W}^{(2)} f_{\mathrm{ReLU}}(\boldsymbol{W}^{(1)}\boldsymbol{X} + \theta^{(1)}) \\ &\qquad + \theta^{(2)})\cdots) + \theta^{(l)} \end{aligned}, \tag{9}$$

where $\boldsymbol{Z}$ represents the linearly weighted output of input matrix $\boldsymbol{X}$ and model weights $\boldsymbol{W}$, as described in Equation (3), and θ represents the offset. The index l denotes the layer of the DNN model, which includes the input layer, hidden layers, and output layer. Our network employs a total of four different DNN models with varying hyperparameters: $\boldsymbol{Y}^f$ for predicting spectral features, $\boldsymbol{Y}^e$ for predicting equalizer parameters, $\boldsymbol{Y}^E$ for predicting the eye diagram, and $\boldsymbol{Y}^C$ for predicting the COM.

The augmented matrix $\boldsymbol{X}_f$ is obtained by horizontally concatenating the input matrix $\boldsymbol{X}$ of the first DNN model with its output $\boldsymbol{Y}^f$:

$$\begin{aligned} \boldsymbol{X}^f &= [\boldsymbol{X} \Big| \boldsymbol{Y}^f] \\ &= \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N_P} & y_{1I} & y_{1R} \\ x_{21} & x_{22} & \cdots & x_{2N_P} & y_{2I} & y_{2R} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ x_{N_C1} & x_{N_C2} & \cdots & x_{N_CN_P} & y_{N_CI} & y_{N_CR} \end{bmatrix} \end{aligned}, \tag{10}$$

where the column vector $\boldsymbol{y}_I = (y_{1I}, y_{2I}, \cdots, y_{N_cI})^T$ represents the predicted channel IL and $\boldsymbol{y}_R = (y_{1R}, y_{2R}, \cdots, y_{N_cR})^T$ represents the predicted channel RL.

To enhance the accuracy of the predicted COM value, the pulse response is extracted from the pre- and post-equalization SBR in the COM tool. The peak corresponding time point is taken as the main cursor, and the amplitudes of four pre-cursors and eight post-cursors determined by the COM sampling interval are selected as the required SBR. Then, the pulse of each channel is flattened and the corresponding time labels are inserted, forming a sequential feature vector. Equation (11) represents the sequence of amplitude values. After this processing, the Transformer model can be employed to accurately predict the SBR.

$$\begin{cases} SBR_1(t) = F_{Trans}[x_{11}(t), x_{12}(t), \cdots, \\ \qquad\qquad x_{1N_P}(t), y_{1I}(t), y_{1R}(t)] \\ \boldsymbol{SBR}_1 = [SBR_1(t - t_0), \cdots, \\ \qquad\qquad SBR_1(t), \cdots, SBR_1(t + t_1)] \end{cases}, \tag{11}$$

where F_{Trans} denotes the Transformer model being utilized, $SBR_1(t)$ represents the SBR corresponding to the input feature vector of the channel at a specific time point, and $\boldsymbol{SBR}_1$ represents the SBR sequence of the channel. t_0 and t_1 are the left and right boundaries of the time range, respectively.

By feeding the SBR into the subsequent DNN model with distinct parameters, the prediction of equalizer parameters can be achieved via Equation (12):

$$\boldsymbol{Y}^e = F_{DNN}(\boldsymbol{SBR}_1, \boldsymbol{SBR}_2, \cdots, \boldsymbol{SBR}_{N_C}) \tag{12}$$

By integrating the equalizer parameters with the input featuring spectral characteristics, an augmented matrix $SBR^{eq}_{N_C}(t)$ is constructed for the prediction of post-equalization

SBR. This matrix, via Equation (13), is then processed into a time-dependent sequence following the methodology presented in Equation (11).

$$\begin{cases} SBR_{N_C}^{eq}(t) = F_{Trans}[x_{N_C1}(t), \cdots, x_{\mathrm{N_C N_P}}(t), \\ \qquad\qquad \mathrm{y_{N_C I}}(t), \mathrm{y_{N_C R}}(t), \boldsymbol{Y}_{\mathrm{N_C}}^{\mathbf{e}}(t)] \\ \boldsymbol{SBR}_{N_C}^{eq} = [SBR_{N_C}^{eq}(t-t_0), \cdots, \\ \qquad\qquad SBR_{N_C}^{eq}(t), \cdots, SBR_{N_C}^{eq}(t+t_1)] \\ \boldsymbol{Y}_{N_C}^{e}(t) = [\boldsymbol{y}_{FFE-N_C}^{Taps}(t), \boldsymbol{y}_{CTLE-N_C}^{DCgain}(t), \boldsymbol{y}_{DFE-N_C}^{Taps}(t)] \end{cases} \tag{13}$$

Finally, the obtained post-equalization SBR is concatenated with $\boldsymbol{X}^f$ to form the final input features for predicting the COM value.

Before formally training the cascaded model, we standardized the input data and employed a pre-training strategy. Initially, the DNN models $F_{DNN}^{freq}(\boldsymbol{X})$ and $F_{DNN}^{eq}(\boldsymbol{X})$ designed for predicting spectral features and equalizer parameters, respectively, were trained independently, and their corresponding parameters were saved. These pre-trained DNN models were then integrated with the necessary Transformer model to construct the complete cascaded model for predicting the equalizer parameters and COM, as depicted in Figure 9. For training the cascaded model, Smooth L1 was used as the cost function and the Adam algorithm was applied to update the model parameters. With the help of the optuna toolkit, model structuring and hyperparameter optimization (HPO) were performed via Bayesian optimization using the GP [37,38]. The chosen acquisition function was the Expected Improvement (EI) function, with an initial set of 10 observation points, allowing for effective HPO within a computationally feasible scope.

Compared to the global grid search, random grid search, and halving search methods, Bayesian optimization is more efficient for HPO and requires less optimization time. Additionally, the K-Fold cross-validation method is employed in this paper to enhance the reliability of the model evaluation.

After completing the model's training, its predictive capabilities can be assessed with several different metrics. One of these metrics, shown in Equation (14), is the RMSE. The squaring operation in RMSE increases its sensitivity to errors, making it particularly responsive to outliers. This increased sensitivity can also make it overly responsive to outliers, but the square-root operation cannot intuitively reflect the actual magnitude of error.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_n - \hat{y}_n)^2} \tag{14}$$

MAE is calculated according to the average value of absolute differences between the predicted and actual values. Compared to RMSE, MAE is less sensitive to outliers and provides a more intuitive reflection of the actual magnitude of errors. However, this can lead to an underestimation of the impact of the errors.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_n - \hat{y}_n| \tag{15}$$

MAPE is the average value of absolute percentage differences between the predicted and actual values. Compared to RMSE and MAE, MAPE is easier to understand and allows for the comparison of prediction results across different scales. It is more suitable for situations with significant changes in actual values. However, when the actual values are close to zero, the error can become large, so MAPE is unsuitable for datasets containing zero or near-zero values.

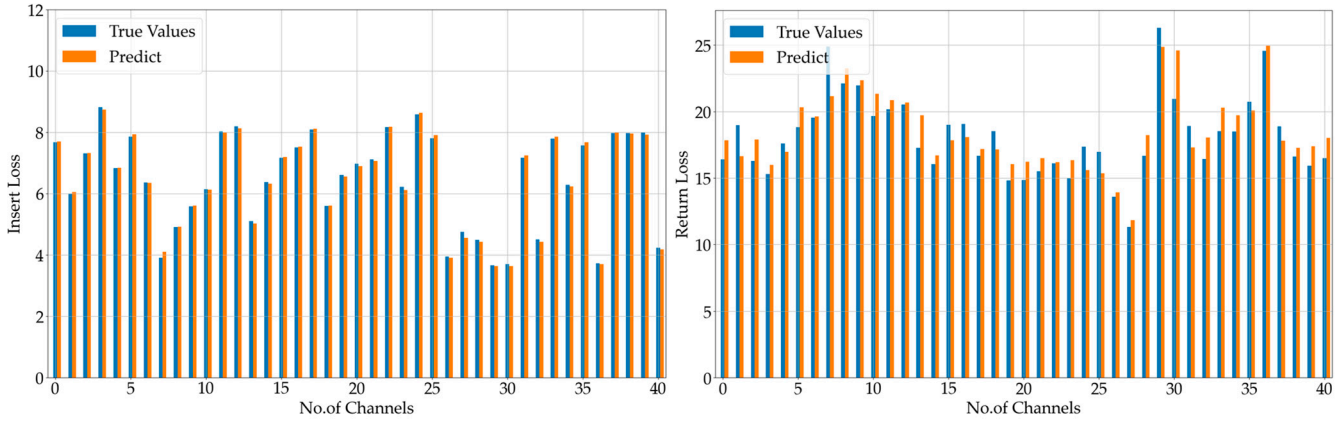
$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_n - \hat{y}_n}{y_n}\right| \tag{16}$$

5. Numerical Results

In this work, the neural network model has been constructed using a Python library based on Pytorch 2.0.1. The computer used for model computation included a GeForce RTX 4070 and an Intel Core i5-12600 KF 3.7 GHz.

5.1. IL and RL

As shown in Figure 10, the IL and RL are losses at Nyquist frequency, and the model's predictions of IL are more accurate than its predictions of RL. This is because IL is primarily influenced by the dielectric loss and input features, including dielectric loss-related parameters. Conversely, RL is affected by additional factors such as impedance mismatches and complex trace shapes, complicating its prediction. Therefore, the accuracy of IL prediction is better than that of RL prediction. The relative error of RL prediction is 17.4%, but this is sufficient to support the accurate prediction of eye diagrams and COM values.

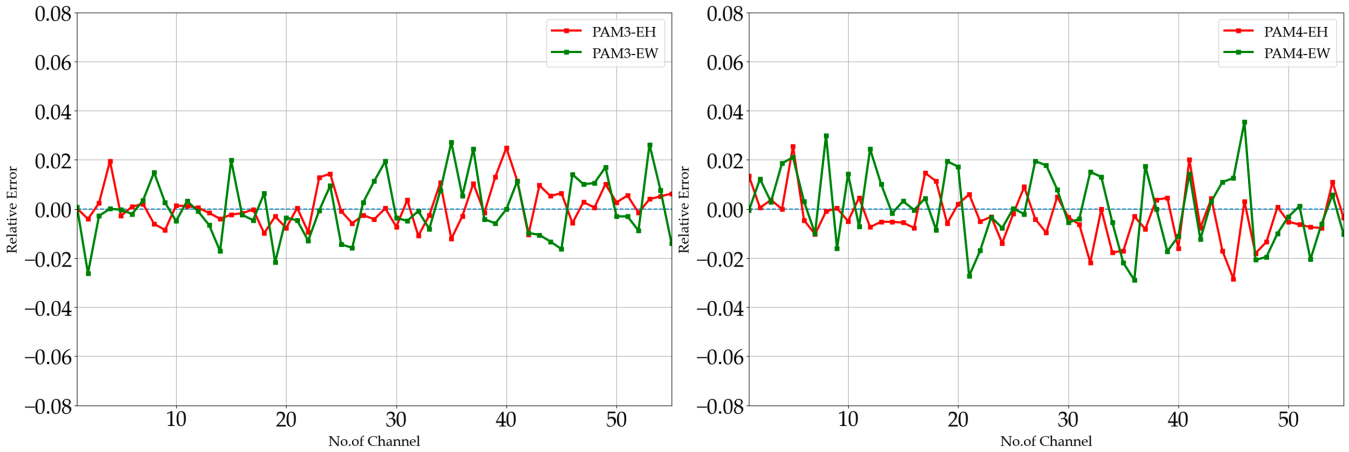


(**a**) (**b**)

Figure 10. Comparison of predicted and actual values for (**a**) IL and (**b**) RL.

5.2. EH/EW of the Eye Diagrams

Figure 11 shows the relative errors in eye height and width for 40 test channels with the PAM3 and PAM4 modulation methods. Channels No. 1 to No. 32 are stripline channels, while channels No. 33 to No. 56 are microstrip channels. Channels No. 1 to No. 10 exhibit more complex trace paths, resulting in larger errors in eye height and eye width. In contrast, channels No. 11 to No. 23, which have simpler routing with straighter or fewer bend paths, show better fitting precision. Most channels in the dataset utilize FR-4 as the dielectric layer, whereas channels No. 24 to No. 32 employ alternative materials, resulting in relatively larger errors. Compared to the stripline channels, channels No. 33 to No. 56, which utilize microstrip lines, show significantly higher relative errors, demonstrating that microstrip lines have weaker anti-interference capabilities than striplines. Table 4 shows the RMSE, MAE, MAPE, and MRE for the eye diagram prediction with the PAM3 and PAM4 modulation methods. It is evident that the prediction results exhibit good convergence and precision. Because the EH unit is chosen as mV, the RMSE and MAE metrics increase by an order of magnitude compared to EW. The results in this table, combined with those in Figure 11, demonstrate that the PAM3 and PAM4 models are effective and exhibit high accuracy.



(**a**) PAM3 (**b**) PAM4

Figure 11. Relative errors of eye diagrams for test set channels in the two modulation formats.

Table 4. Prediction of eye diagrams with the DNN model.

Eye	Modulation	RMSE	MAE	MAPE	MRE (%)
EH (mV)	PAM3	4.8×10^{-1}	3.3×10^{-1}	5.7×10^{-3}	2.5
	PAM4	4.0×10^{-1}	2.9×10^{-1}	8.1×10^{-3}	2.8
EW (UI)	PAM3	4.4×10^{-3}	3.4×10^{-3}	9.3×10^{-3}	2.7
	PAM4	4.4×10^{-3}	3.6×10^{-3}	1.2×10^{-2}	3.5

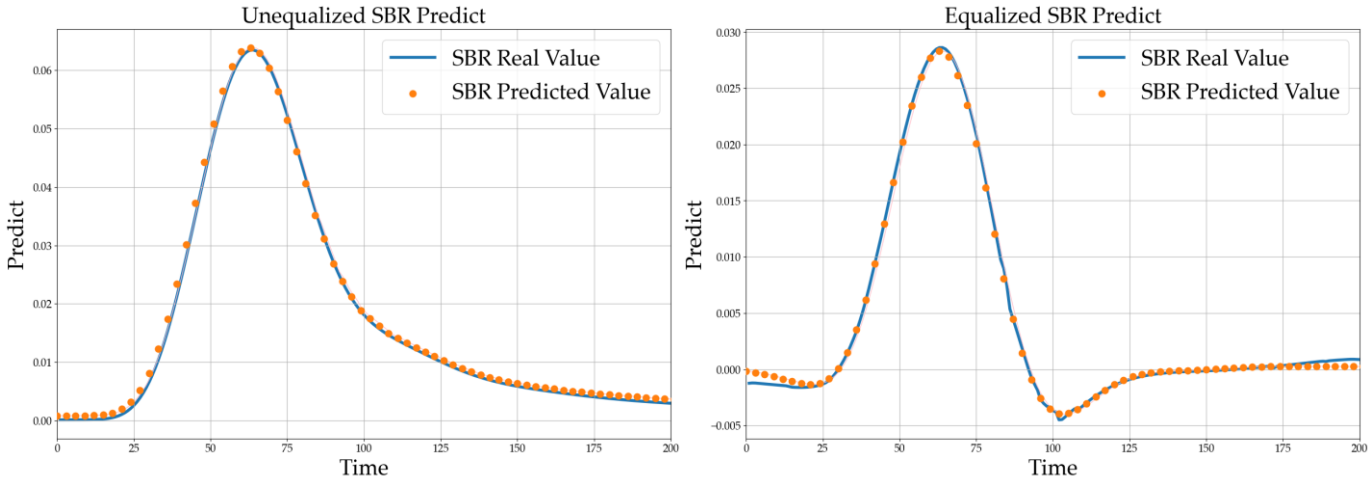
5.3. COM

Because many pre-cursor and post-cursor SBR values are close or equal to zero, MAPE and MRE are not suitable for evaluating the accuracy of SBR predictions, and only RMSE and MAE are used to evaluate the prediction accuracy for the SBR. For the SBR before equalization of the 40 test channels, the RMSE is 7.3×10^{-4} and the MAE is 6.3×10^{-4}. For the SBR after equalization, the RMSE is 1.1×10^{-3} and the MAE is 1.6×10^{-4}. Below, we use a randomly selected channel (ID: 3) from the test set to illustrate the SBR prediction performance. As depicted in Figure 12, both the pre-equalization and post-equalization SBR can be predicted by the Transformer model with high precision.

For the PAM4 signal format, Figure 13 provides a concise demonstration. We selected an equalizer configuration that includes a 3-taps TX FFE, an RX CTLE with dual DC gain, and a 12-taps RX DFE (hereafter referred to as the standard equalizer configuration). As shown in Figure 13, channels No. 7 and No. 30 have large relative errors due to their complex shapes. Some channels with high RL, such as channels No. 7–No. 9, also exhibit significant relative error. The relative error of the predicted COM values for each test channel is within 2%, demonstrating the ability of the DNN–Transformer model to achieve the required prediction accuracy.

Here, the influences of different combinations of equalizers on the prediction precision of COM values are analyzed deeply. As shown in Table 5, we configured five different equalizer combinations. The results indicate that our proposed cascaded model achieves the expected prediction performance across various equalizer combinations. However, it is apparent that a reduction in the variety of equalizers weakens the model's generalization capability and decreases its prediction accuracy. When only the DFE was enabled, the RMSE increased to 1.6×10^{-1}, and the MRE reached 17.5%. We attribute this to the optimization method of this equalizer within the COM framework and the fact that the cascaded model also takes into account predictions for both the CTLE and FFE. Therefore, we adjusted the network structure by removing unused equalizers in the cascaded model for different

equalizer combinations. This modification has resulted in an improvement in the model's prediction accuracy for COM values, with the MRE decreasing from 17.5% to 11.5%.



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

Figure 12. True and predicted results for (**a**) the SBR before equalization and (**b**) the SBR after equalization for 3-taps FFE + CTLE + 12-taps DFE test set channels (ID: 3).

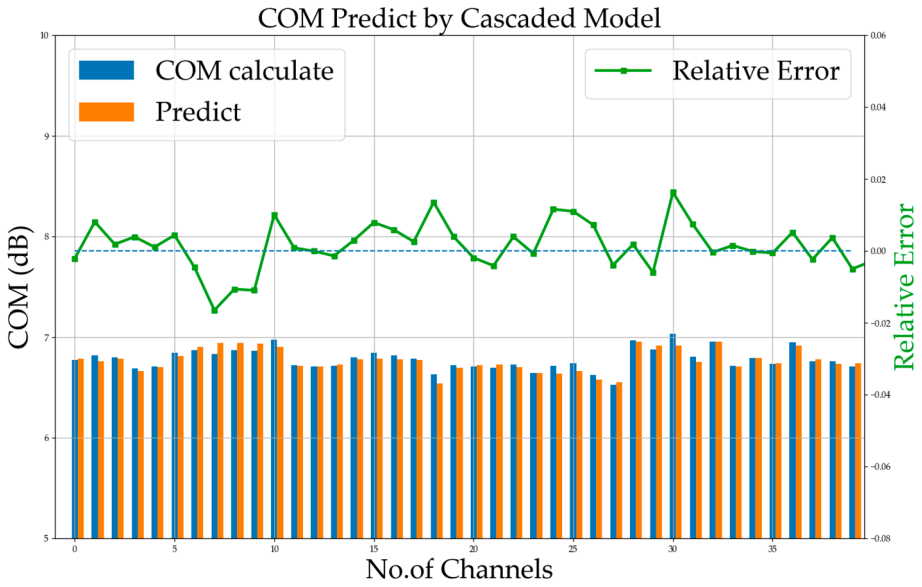
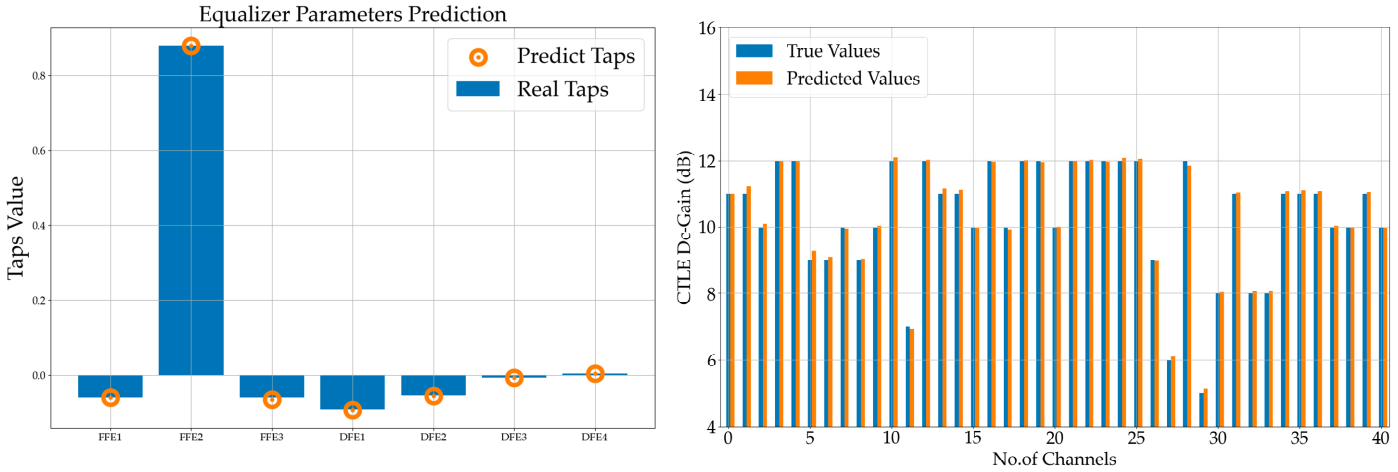


Figure 13. Relative error and COM values for PAM4 coding (ID: 1–40) in the standard equalizer configuration.

Table 5. Predicted COM error values for the cascaded model.

COM	RMSE	MAE	MAPE	MRE (%)
3-taps FFE + CTLE + 12-taps DFE	4.5×10^{-2}	3.5×10^{-2}	5.1×10^{-3}	1.6%
3-taps FFE + CTLE + 8-taps DFE	5.0×10^{-2}	3.7×10^{-2}	5.6×10^{-3}	2.0%
3-taps FFE + CTLE + 4-taps DFE	4.2×10^{-2}	3.2×10^{-2}	4.1×10^{-3}	1.7%
CTLE + 12-taps DFE	9.6×10^{-2}	7.1×10^{-2}	1.4×10^{-2}	5.7%
12-taps DFE	1.6×10^{-1}	1.1×10^{-1}	3.0×10^{-2}	17.5%

Below, a test channel (ID: 3) is used to illustrate the prediction capability of the model with the first four taps of the DFE and three taps of the FFE. As shown in Figure 14a, the cascaded model realizes highly accurate predictions. Furthermore, the prediction results for the main DC gain of the CTLE for the 40 test channels, as presented in Figure 14b, also indicate a high level of accuracy.



(**a**) FFE and DFE taps (**b**) CTLE DC gain

Figure 14. Comparison of predicted and actual equalizer parameter results.

The predictive capabilities of different cascaded model combinations were investigated as follows. The equalizer configuration was set as the standard configuration, and the HPO parameters were uniformly configured, with the learning rate search range set from 0.0001 to 0.01, conducting 20 trials for optimization. Given the characteristics of the SBR, we tested the predictive performance of standalone models, including a DNN, an LSTM network, and a Transformer. The DNN combined with two sequence prediction models, LSTM and a Transformer, was selected as a cascaded scheme. The DNN-LSTM model replaces the Transformer module in Figure 9 with a typical LSTM network. The DNN, LSTM, and Transformer use independent models without time label processing modules, so these models can only achieve eye diagram and COM value predictions. The best structural parameters of these models—such as the number of hidden layers and the hidden layer size of the LSTM, and the head number, vector dimension, and feedforward network dimension of the Transformer's attention mechanism—were all obtained through Bayesian optimization. As presented in Table 6, the cascaded models demonstrate superior predictive capabilities compared to the standalone models. Among the cascaded models, those incorporating a Transformer display the best predictive performance, indicating that the DNN–Transformer cascaded model is the optimal COM prediction model.

Table 6. Predicted error values of different models.

Model	RMSE	MAE	MAPE	MRE (%)
DNN	6.9×10^{-2}	4.3×10^{-2}	7.8×10^{-3}	6.5
LSTM	5.3×10^{-2}	4.4×10^{-2}	6.4×10^{-3}	2.0
DNN-LSTM	5.0×10^{-2}	4.1×10^{-2}	6.0×10^{-3}	1.8
Transformer	4.7×10^{-2}	3.7×10^{-2}	5.5×10^{-3}	1.7
DNN–Transformer	4.6×10^{-2}	3.5×10^{-2}	5.1×10^{-3}	1.6

As shown in Table 7, we have compared the DNN–Transformer model with the models presented in the Introduction section in terms of functionality. The traditional LSTM [20]

structure could not achieve prediction under physical parameter variables. Traditional machine learning models such as the SVM [13], RFR [12], and DNN [9] models have limited processing capabilities for sequences and thus cannot predict transient waveforms. The GNN-RNN [21] model simplifies the interconnected components and circuits, meaning that some of the actual parameters of links are lost and the performance is decreased. The above models could not achieve link optimization, and RFR [12] could only complete FD prediction. In contrast, the proposed DNN–Transformer is more practical. This model can effectively achieve these above behavioral tasks and performs better than other models in RMSE and MAE for COM value prediction. Due to the increased complexity of this model, its inference time is increased, but it still has a significant time advantage compared to traditional EM simulation whose single-solution time is several hours.

Table 7. Functional comparison of different models for high-speed links.

Model	Physical Parameters Variable	Transient Waveform Prediction	FD Prediction	Link Optimization	GPU/CPU Time (ms)	RMSE	MAE
SVM [13]	Yes	No	No	No	<5 *	3.8×10^{-1}	3.1×10^{-1}
RFR [12]	Yes	No	Yes	No	<5 *	4.8×10^{-1}	3.8×10^{-1}
DNN [9]	Yes	No	No	No	<5	6.9×10^{-2}	4.3×10^{-2}
LSTM [20]	No	Yes	No	No	<5	5.3×10^{-2}	4.4×10^{-2}
GNN-RNN [21]	Yes	Yes	No	No	567.1	\	\
DNN–Transformer	Yes	Yes	Yes	Yes	792.8 *	4.6×10^{-2}	3.5×10^{-2}

* GPU inference time.

6. Conclusions

In this paper, a DNN–Transformer cascaded neural network is proposed for effectively analyzing the SI in high-speed serial links. During the dataset creation process, we referenced the USB4 Gen4 and 50GBASE-KR standards for PCB design and electromagnetic simulation, and used the physical design parameters of each channel as inputs for the model. This DNN–Transformer model is used in this paper to extract the features from physical design parameters of channels and successfully predict the eye diagram data and COM values of test links. In addition, this deep learning model can successfully predict the SBR before and after equalization, and main equalizer parameters for different combinations are also accurately predicted. For the model's training, we employed a Bayesian optimization method based on the GP for HPO. Finally, this paper compares DNN–Transformer with various other models such as DNN, LSTM, Transformer, and DNN-LSTM models. The results shows that our DNN–Transformer cascaded model accurately achieves performance prediction and equalization architecture optimization for high-speed serial links, and the MRE in its COM prediction results for the test set, with an equalizer configuration comprising a 3-taps TX FFE, an RX CTLE with dual DC gain, and a 12-taps RX DFE, is 1.6%.

Author Contributions: Conceptualization, L.W., J.Z. and Y.Z. (Yinhang Zhang); methodology, L.W., J.Z. and H.J.; software, L.W.; validation, L.W., J.Z. and Y.Z. (Yinhang Zhang); formal analysis, L.W., H.J. and Y.Z. (Yinhang Zhang); investigation, L.W.; resources, L.W., Y.Z. (Yinhang Zhang) and Y.Z. (Yongzheng Zhan); data curation, L.W. and Y.Z. (Yinhang Zhang); writing—original draft preparation, L.W.; writing—review and editing, X.Y., Y.Z. (Yinhang Zhang) and Y.Z. (Yongzheng Zhan); visualization, L.W.; supervision, X.Y., Y.Z. (Yinhang Zhang) and Y.Z. (Yongzheng Zhan); project administration, L.W.; funding acquisition, L.W., Y.Z. (Yinhang Zhang) and X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 61861019, 62161012), the Hunan Provincial Department of Education (Grant No. 22B0525, 21A0335),

the China Postdoctoral Science Foundation (Grant No. 2024M751268), and the Postgraduate Research Program of Jishou University (Grant No. Jdy23035).

Data Availability Statement: The data used to support the findings of the study are available within the article.

Conflicts of Interest: Author Yongzheng Zhan was employed by the company Shandong Yunhai Guochuang Cloud Computing Equipment Industry Innovation Company Limited. 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.

Abbreviations

The following abbreviations are used in this manuscript:

DNN	Deep Neural Network;
LSTM	Long Short-Term Memory Neural Network;
SBR	Single-Bit Response;
SI	signal integrity
COM	channel operating margin;
eCOM	enhanced channel operating margin;
GP	Gaussian process;
HPO	hyperparameter optimization;
EM	electromagnetic;
EMFS	electromagnetic field solver;
IBIS-AMI	Input/Output Buffer Information Specification Algorithmic Model Interface;
EH	eye height;
EW	eye width;
RL	return loss;
IL	insertion loss;
TD	time domain;
FD	frequency domain;
ML	machine learning;
SVM	Support Vector Machine;
LS-SVM	Least-Squares Support Vector Machine;
FNN	Feedforward Neural Network;
RFR	Random Forest Regression;
RNN	Recurrent Neural Network;
FFE	feedforward equalizer;
DFE	decision feedback equalizer;
CTLE	continuous-time linear equalizer;
ILD	insertion loss deviation;
ICR	insertion loss-to-crosstalk ratio;
ISI	inter-symbol interference;
FOM	figure of merit.

References

1. Hall, S.H.; Heck, H.L. *Advanced Signal Integrity for High-Speed Digital Designs*; John Wiley & Sons: Hoboken, NJ, USA, 2009; pp. 201–206.
2. Yan, J.; Zargaran-Yazd, A. IBIS-AMI modelling of high-speed memory interfaces. In Proceedings of the 2015 IEEE 24th Electrical Performance of Electronic Packaging and Systems (EPEPS), San Jose, CA, USA, 25–28 October 2015; pp. 73–76.
3. LAN/MAN Standards Committee of the IEEE Computer Society. IEEE Standard for Ethernet. 2012. Available online: https://ieeexplore.ieee.org/document/6419735 (accessed on 26 June 2024).
4. Wang, Y.; Hu, Q.S. A COM Based High Speed Serial Link Optimization Using Machine Learning. *IEICE Trans. Electron.* **2022**, *105*, 684–691. [CrossRef]
5. Gore, B.; Richard, M. An excerise in applying channel operating margin (COM) for 10GBASE-KR channel design. In Proceedings of the 2014 IEEE International Symposium on Electromagnetic Compatibility (EMC), Raleigh, NC, USA, 4–8 August 2014; pp. 653–658.

6. Ambasana, N.; Gope, B.; Mutnury, B.; Anand, G. Application of artificial neural networks for eye-height/width prediction from S-parameters. In Proceedings of the IEEE 23rd Conference on Electrical Performance of Electronic Packaging and Systems, Portland, OR, USA, 26–29 October 2014; pp. 99–102.
7. Ambasana, N.; Anand, G.; Mutnury, B.; Gope, D. Eye Height/Width Prediction From S-Parameters Using Learning-Based Models. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2016**, *6*, 873–885. [CrossRef]
8. Ambasana, N.; Anand, G.; Gope, D.; Mutnury, B. S-Parameter and Frequency Identification Method for ANN-Based Eye-Height/Width Prediction. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2017**, *7*, 698–709. [CrossRef]
9. Lu, T.; Sun, J.; Wu, K.; Yang, Z. High-Speed Channel Modeling with Machine Learning Methods for Signal Integrity Analysis. *IEEE Trans. Electromagn. Compat.* **2018**, *60*, 1957–1964. [CrossRef]
10. Lho, D.; Park, J.; Park, H.; Kang, H.; Park, S.; Kim, J. Eye-width and Eye-height Estimation Method based on Artificial Neural Network (ANN) for USB 3.0. In Proceedings of the 2018 IEEE 27th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS), San Jose, CA, USA, 14–17 October 2018; pp. 209–211.
11. Sánchez-Masís, A.; Rimolo-Donadio, R.; Roy, K.; Sulaiman, M.; Schuster, C. FNNs Models for Regression of S-Parameters in Multilayer Interconnects with Different Electrical Lengths. In Proceedings of the 2023 IEEE MTT-S Latin America Microwave Conference (LAMC), San Jose, CA, USA, 6–8 December 2023; pp. 82–85.
12. Li, X.; Hu, Q. A Machine Learning based Channel Modeling for High-speed Serial Link. In Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC), Chengdu, China, 11–14 December 2020; pp. 1511–1515.
13. Trinchero, R.; Canavero, F.G. Modeling of eye diagram height in high-speed links via support vector machine. In Proceedings of the 2018 IEEE 22nd Workshop on Signal and Power Integrity (SPI), Brest, France, 22–25 May 2018; pp. 1–4.
14. Cao, Y.; Zhang, Q.J. A New Training Approach for Robust Recurrent Neural-Network Modeling of Nonlinear Circuits. *IEEE Trans. Microw. Theory Tech.* **2009**, *57*, 1539–1553. [CrossRef]
15. Mutnury, B.; Swaminathan, M.; Libous, J.P. Macromodeling of Nonlinear Digital I/O Drivers. *IEEE Trans. Adv. Packag.* **2006**, *29*, 102–113. [CrossRef]
16. Cao, Y.; Erdin, I.; Zhang, Q.J. Transient Behavioral Modeling of Nonlinear I/O Drivers Combining Neural Networks and Equivalent Circuits. *IEEE Microw. Wirel. Compon. Lett.* **2010**, *20*, 645–647. [CrossRef]
17. Yu, H.; Michalka, T.; Larbi, M.; Swaminathan, M. Behavioral Modeling of Tunable I/O Drivers with Preemphasis Including Power Supply Noise. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2020**, *28*, 233–242. [CrossRef]
18. Yu, H.; Chalamalasetty, H.; Swaminathan, M. Modeling of Voltage-Controlled Oscillators Including I/O Behavior Using Augmented Neural Networks. *IEEE Access* **2019**, *7*, 38973–38982. [CrossRef]
19. Faraji, A.; Noohi, M.; Sadrossadat, S.A.; Mirvakili, A.; Na, W.C.; Feng, F. Batch-Normalized Deep Recurrent Neural Network for High-Speed Nonlinear Circuit Macromodeling. *IEEE Trans. Microw. Theory Tech.* **2022**, *70*, 4857–4868. [CrossRef]
20. Moradi, M.; Sadrossadat, A.; Derhami, V. Long Short-Term Memory Neural Networks for Modeling Nonlinear Electronic Components. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2021**, *1*, 840–847. [CrossRef]
21. Li, Z.; Li, C.X.; Wu, Z.M.; Zhu, Y.; Mao, J.F. Surrogate Modeling of High-Speed Links Based on GNN and RNN for Signal Integrity Applications. *IEEE Trans. Microw. Theory Tech.* **2023**, *71*, 3784–7796. [CrossRef]
22. Lho, D.; Park, H.; Park, S.; Kim, S.; Kang, H.; Sim, B.; Kim, S.; Park, J.; Cho, K.; Song, J.; et al. Channel Characteristic-Based Deep Neural Network Models for Accurate Eye Diagram Estimation in High Bandwidth Memory (HBM) Silicon Interposer. *IEEE Trans. Electromagn. Compat.* **2021**, *64*, 196–208. [CrossRef]
23. Li, G.S.; Mao, C.S.; Zhao, W.S. Semi-supervised Regression Model for Eye Diagram Estimation of High Bandwidth Memory (HBM) Silicon Interposer. In Proceedings of the 2023 International Applied Computational Electromagnetices Society Symposium, Hangzhou, China, 15–18 August 2023; pp. 1–3.
24. Goay, C.H.; Ahmad, N.S.; Goh, P. Temporal Convolutional Networks for Transient Simulation of High-Speed Channels. *Alex. Eng. J.* **2023**, *74*, 643–663. [CrossRef]
25. Li, B.W.; Jiao, B.; Chou, C.H.; Mayder, R.; Franzon, P. Self-Evolution Cascade Deep Learning Model for High-Speed Receiver Adaptation. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2020**, *10*, 1043–1053. [CrossRef]
26. Li, B.W.; Jiao, B.; Chou, C.H.; Mayder, R.; Franzon, P. CTLE Adaptation Using Deep Learning in High-Speed SerDes Link. In Proceedings of the IEEE 70th Electronic Components and Technology Conference (ECTC), Orlando, FL, USA, 3–30 June 2020; pp. 952–955.
27. Zhang, H.H.; Xue, Z.S.; Liu, X.Y.; Li, P.; Jiang, L.J.; Shi, G.M. Optimization of High-Speed Channel for Signal Integrity with Deep Genetic Algorithm. *IEEE Trans. Electromagn. Compat.* **2022**, *64*, 1270–1274. [CrossRef]
28. Shan, G.; Li, G.; Wang, Y.; Xing, C.; Zheng, Y.; Yang, Y. Application and Prospect of Artificial Intelligence Methods in Signal Integrity Prediction and Optimization of Microsystems. *Micromachines* **2023**, *14*, 344. [CrossRef]
29. Mellitz, R.; Ran, A.; Li, M.P.; Ragavassamy, V. Channel Operating Margin (COM): Evolution of Channel Specifications for 25 Gbps and Beyond. In Proceedings of the DesignCon 2013, Santa Clara, CA, USA, 28–31 January 2013.
30. Pu, B.; He, J.; Harmon, A.; Guo, Y.; Liu, Y.; Cai, Q. Signal Integrity Design Methodology for Package in Co-packaged Optics Based on Figure of Merit as Channel Operating Margin. In Proceedings of the 2021 IEEE International Joint EMC/SI/PI and EMC Europe Symposium, Raleigh, NC, USA, 26 July–13 August 2021; pp. 492–497.
31. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [CrossRef]

32. Bono, F.M.; Radicioni, L.; Cinquemani, S. A Novel Approach for Quality Control of Automated Production Lines Working under Highly Inconsistent Conditions. *Eng. Appl. Artif. Intell.* **2023**, *122*, 106149. [CrossRef]
33. Steel, R.G.D.; Torrie, J.H. *Principles and Procedures of Statistics*; McGraw Hill: New York, NJ, USA, 1960.
34. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
35. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
36. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
37. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, 4–8 August 2019.
38. Frazier, P.I. A Tutorial on Bayesian Optimization. *arXiv* **2018**, arXiv:1807.02811.

electronics

MDPI

Article

Enhancing Autonomous Vehicle Perception in Adverse Weather: A Multi Objectives Model for Integrated Weather Classification and Object Detection

Nasser Aloufi *, Abdulaziz Alnori and Abdullah Basuhail

Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; asalnori@kau.edu.sa (A.A.); abasuhail@kau.edu.sa (A.B.)
* Correspondence: nhassanaloufi@stu.kau.edu.sa

Abstract: Robust object detection and weather classification are essential for the safe operation of autonomous vehicles (AVs) in adverse weather conditions. While existing research often treats these tasks separately, this paper proposes a novel multi objectives model that treats weather classification and object detection as a single problem using only the AV camera sensing system. Our model offers enhanced efficiency and potential performance gains by integrating image quality assessment, Super-Resolution Generative Adversarial Network (SRGAN), and a modified version of You Only Look Once (YOLO) version 5. Additionally, by leveraging the challenging Detection in Adverse Weather Nature (DAWN) dataset, which includes four types of severe weather conditions, including the often-overlooked sandy weather, we have conducted several augmentation techniques, resulting in a significant expansion of the dataset from 1027 images to 2046 images. Furthermore, we optimize the YOLO architecture for robust detection of six object classes (car, cyclist, pedestrian, motorcycle, bus, truck) across adverse weather scenarios. Comprehensive experiments demonstrate the effectiveness of our approach, achieving a mean average precision (mAP) of 74.6%, underscoring the potential of this multi objectives model to significantly advance the perception capabilities of autonomous vehicles' cameras in challenging environments.

Keywords: autonomous vehicles; conventional neural network; object detection; deep learning; camera sensors; adverse weather; weather classification

Citation: Aloufi, N.; Alnori, A.; Basuhail, A. Enhancing Autonomous Vehicle Perception in Adverse Weather: A Multi Objectives Model for Integrated Weather Classification and Object Detection. *Electronics* **2024**, *13*, 3063. https://doi.org/10.3390/electronics13153063

Academic Editors: Shiping Wen and Jianying Xiao

Received: 14 May 2024
Revised: 22 July 2024
Accepted: 29 July 2024
Published: 2 August 2024

1. Introduction

The rapid advancement of autonomous vehicle (AV) technology has captured the attention of researchers, engineers, policymakers, and the public. Central to AV development are sensors that enable perception and decision-making within dynamic driving environments. Among these, camera sensors play a vital role as the primary source of visual perception in the AV systems. Cameras capture real-time high-resolution images of the vehicle's surroundings, providing crucial visual data for the accurate detection and classification of various objects. By leveraging advanced object detection algorithms, cameras contribute to various AV functionalities such as lane keeping and path planning by continuously monitoring lane markings and changes in road layout. This enables the vehicle to maintain its position within lanes and make informed decisions regarding trajectory and maneuvering, thereby enhancing overall road safety and traffic flow. Furthermore, camera sensors contribute to path planning by identifying obstacles, traffic signs, and other entities, enabling the vehicle to adapt its trajectory accordingly and navigate complex traffic scenarios.

Depth estimation is another key capability of camera sensors, enabling AVs to perceive the distances of surrounding objects accurately. Through advanced image processing techniques, cameras can provide depth perception, enhancing the vehicle's spatial awareness and obstacle avoidance capabilities. Image segmentation is an additional task performed

by camera sensors, wherein the visual scene is segmented into semantically meaningful regions. This segmentation enables the AV to distinguish between various elements within its field of view, facilitating robust object detection and classification, essential for safe navigation. Moreover, camera sensors play a crucial role in the fusion of perception, integrating data from multiple cameras positioned around the vehicle to construct a comprehensive situational awareness map. This fusion enhances the vehicle's understanding of its surroundings, enabling it to make informed decisions in real time. In addition to external perception, camera sensors also contribute to cabin monitoring and provide essential data for passenger behaviors and status. Figure 1 lists the main roles of cameras and their systems in AVs.

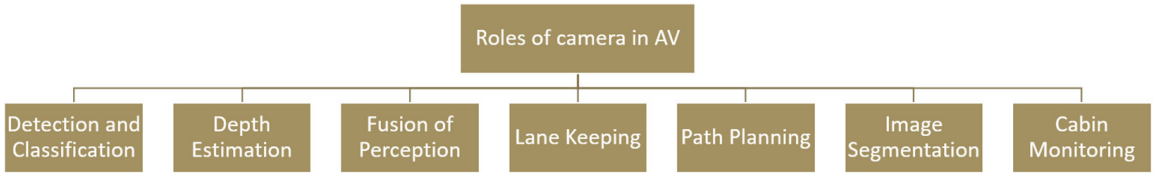


Figure 1. Main roles of cameras and their systems in AVs.

In addition to their primary functions, cameras offer a cost-effective and lightweight solution compared to alternative sensor technologies such as LiDAR and radar. This affordability facilitates widespread adoption and deployment of AV technology, paving the way for a future where autonomous vehicles are ubiquitous on our roads.

Having said that about the functions of cameras, reaching the highest level of AV development, "full automation" [1], requires vehicles to detect every object in their surrounding environment under all conditions and scenarios with no less than human-like behavior. Achieving this level is still very challenging during adverse weather, which presents significant challenges for camera sensors, impacting their abilities to capture clear and reliable images of the environment. Weather conditions such as rain, snow, fog, and sandstorms pose significant challenges for camera sensors, and the key challenges include:

- Reduced visibility: adverse weather conditions often lead to reduced visibility, impairing the effectiveness of camera sensors in capturing clear images of the environment. Rain, snow, and fog can obscure the field of view, making it challenging for cameras to discern objects and obstacles accurately.
- Water droplets and snow accumulation: rain and snow can result in water droplets or snow accumulation on camera lenses, leading to distortion, blurring, or occlusion of captured images. This accumulation can degrade image quality and hinder object detection and recognition capabilities.
- Fog and haze: foggy conditions create a hazy atmosphere that reduces contrast and clarity in camera images, hindering object detection and localization. The presence of fog and haze makes it difficult for cameras to distinguish objects from their background, compromising the reliability of AV perception systems.
- Glare and reflections: glare from wet road surfaces or reflective surfaces can cause reflections in camera images, resulting in overexposed or washed-out images. Glare and reflections can obscure important visual information, making it challenging for AVs to navigate safely in adverse weather conditions.
- Sand and dust particles: sandstorms and dusty conditions can lead to the accumulation of sand and dust particles on camera lenses, obstructing the field of view and degrading image quality. This accumulation of particles can compromise the performance of camera sensors, affecting the reliability of AV perception systems.
- Dynamic lighting conditions: adverse weather conditions can cause rapid changes in lighting conditions, including variations in brightness, contrast, and color temperature. Camera systems must adapt to these dynamic lighting conditions to maintain accurate perception of the environment and ensure reliable object detection and recognition.

- Sensor calibration: adverse weather conditions may necessitate adjustments to camera calibration parameters to compensate for changes in lighting, visibility, and sensor performance. Ensuring accurate sensor calibration is essential for maintaining the reliability and effectiveness of camera sensors in adverse weather conditions.
- Reliability and robustness: adverse weather conditions pose reliability and robustness challenges for camera sensors, requiring them to continue functioning effectively in harsh environmental conditions. Ensuring the durability and resilience of camera sensors is crucial for the safe and reliable operation of AVs in adverse weather.

Figure 2 shows some object detection challenges in adverse weather. Each type of weather has its own obstacles. For instance, in heavy snow and sandstorms, the road boundaries can be obscured by snow and sand. During rain, water droplets on camera lenses lead to distortion and blurring of captured images. All of the abovementioned challenges make it difficult for the camera to accurately perceive objects.

Figure 2. Some adverse weather challenges. (**a**,**b**) Heavy snow can obscure road lanes. (**c**) Rain leads to distortion and blurring of captured images. (**d**,**e**) A sandstorm can obscure road boundaries and changed the lighting of the scene. (**f**) Fog makes it difficult for cameras to distinguish objects from their backgrounds.

When exploring object detection using convolutional neural networks (CNNs), we encounter two primary approaches: one-stage and two-stage. The two-stage approach, pioneered by the introduction of the Region-Based CNN (R-CNN) model in 2014, involves a region proposal stage to identify regions containing objects, followed by feature extraction and object classification [2]. However, this method was slow due to processing each proposed region separately. Fast R-CNN, introduced the following year, improved the speed by passing the entire image through the CNN, generating a feature map for object detection [3]. Faster R-CNN [4] was then introduced to further enhance the performance. A significant advancement occurred in 2017 with the introduction of Mask R-CNN [5]. Mask R-CNN adopts the Feature Pyramid Network (FPN) as its backbone [6] and introduces a novel phase to the detection process by generating a segmentation mask for each object.

On the other hand, the one-stage approach, first demonstrated by Redmon et al. [7] with the YOLO model, encapsulates the entire detection process in a single pass through the CNN. YOLOv2 and subsequent iterations like YOLOv3 [8] introduced a new backbone named Darknet-53 to the architecture. A new version of YOLO called YOLOv4 then proposed aiming at improving both accuracy and speed and achieved notable improvements over its predecessors. YOLOv5, YOLOv7 [9], and YOLOv8 were later iterations of YOLO. Another model called the Single-Shot multibox Detector (SSD), proposed in [10], has achieved competitive results on the VOC2007 dataset, with improvements of the mAP.

In this paper, our aim is to propose a solution for AVs based on camera sensors that can not only detect objects, but also classify weather based on the condition of the scene. The scope of our paper is shown in Figure 3.

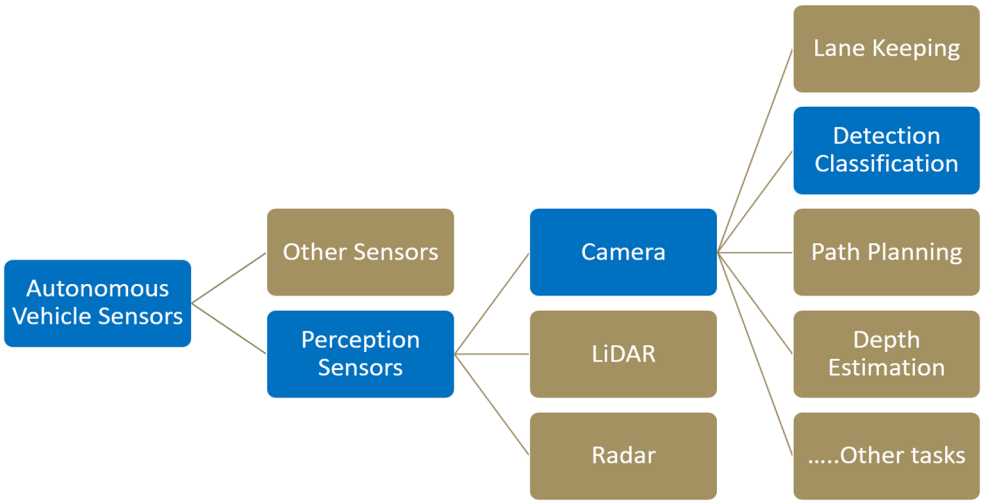


Figure 3. The scope of this paper is highlighted in blue.

The main contributions of our work are as follows:

- We propose a multi objectives model for classifying weather and detecting objects. As we will demonstrate in Section 2, and to the best of our knowledge, existing AV papers treat classifying weather and detecting objects as separate problems. Our proposed model treats road weather classification and road object detection as a unified problem for camera sensing systems.
- We have expanded the Detection in Adverse Weather Nature (DAWN) dataset by adding augmented images that cover all four types of weather conditions (sandy, rainy, foggy, and snowy). The total dataset size has nearly doubled, increasing from its original size.
- In addition to providing a single model for classifying weather and detecting objects, we address a critical gap in autonomous vehicle research by considering sandy weather conditions, which have been largely overlooked by existing studies.
- The base architecture of the object detection model You Only Look Once (YOLO) version 5 has been adapted and modified to suit our domain. As a result, we have successfully increased the mean average precision (mAP) to 74.6%, which is a promising result compared to other papers that used the same dataset.

2. Related Work

Detecting objects in challenging weather conditions presents difficulties because the quality of images degrades and visual features are compromised due to weather phenomena like rain, fog, snow, and sandstorms. These conditions impact detection performance by diminishing scene lighting, reducing object visibility, and complicating object differentiation from surrounding elements. Several papers have been published aiming to propose suitable solutions.

In [11], the authors studied adverse weather classification along with light level in the AV environment. To tackle the issue of perception under adverse weather and low light conditions, where accuracy degradation is a significant concern, the authors introduced their own dataset. The dataset was designed to cover three types of weather (fog, rain, and snow) and three levels of lighting (bright, moderate, and low), along with three street types (asphalt, grass, and cobblestone). The processed images of the dataset contain three labels related to the type of weather, lighting level, and street type. The authors used ResNet18 as their backbone and concluded that the system performed with low accuracy on the dataset and needed further enhancement.

In [12], the authors address the challenges of AVs during adverse weather conditions, where typical perceptual models struggle. Existing research mainly focuses on classifying weather conditions; however, the authors studied the transitions between these types of

weather. They proposed a method to define and understand six intermediate weather transition states (cloudy to rainy, rainy to cloudy, sunny to rainy, rainy to sunny, sunny to foggy, and foggy to sunny). The approach involves interpolating intermediate weather transition data using a variation autoencoder, extracting spatial features with VGG (Visual Geometry Group) very deep convolutional networks, and modeling temporal distribution with a gated recurrent unit for classification. The authors proposed a new large-scale dataset called AIWD6 (Adverse Intermediate Weather Driving), and the results showed an effective weather transition model.

In [13], the authors introduces a novel framework called WeatherNet, which employs four deep CNN models based on the ResNet50 architecture. WeatherNet autonomously extracts weather information from the input image and classifies the output into the right category. However, the drawback of the presented framework is the inability to share features, since the four models work separately.

Ref. [14] focuses on the significant impact of adverse weather conditions on urban traffic and highlights the importance of weather condition recognition for applications such as AV assistance and intelligent transportation systems. Leveraging advancements in deep learning, the paper introduces a new simplified model called ResNet15, a proposed version of the famous ResNet50 [15]. The proposed model has a fully connected layer that used the Softmax classifier. The paper also introduces a new dataset called "WeatherDataset-4" containing about 5000 images covering foggy, rainy, snowy, and sunny weather. Although the proposed network outperformed the traditional ResNet50, the paper lacks coverage of nighttime and sandy environments.

In [16], the authors proposed the MCS-YOLO algorithm to enhance object detection by integrating a coordinate attention mechanism, a multiscale structure for small objects, and applying the Swin Transformer structure [17]. Through experiments on the BDD100K dataset, they demonstrated a mean average precision (mAP) of 53.6%.

Paper [18] is one of the earliest papers that applied CNN for AV weather classification. The authors added two fully connected layers to extract features from Road Service Conditions (RSC) images. The paper focused on winter road conditions, where the problem of snowy roads was divided into three experiments: (a) two-class classification, (b) three-class classification, and (c) five-class classification. The model surpassed traditional classification techniques and recorded an accuracy of 78.5% when applying five-class classification. In [19], YOLOv4 has been enhanced to detect objects through proposing an anchor-free and decoupled head. The paper used BDD100k as original dataset and created a new version that focuses on three types of weather (rainy, snow, foggy). The experimental results showed a mAP of 60.3%.

In [20], the authors extracted high-precision motion data and proposed a new vehicle tracking mechanism called SORT++. Image-Adaptive YOLO (IA-YOLO) was presented in [21] and showed an improvement in detecting objects in low light and foggy environments.

Ref. [22] proposed Dual Subnet Network (DSNet) for detecting objects and achieved a mAP of 50.8% of foggy weather. In [23] YOLOv5 was investigated to detect objects of several classes, and the mAP of all classes scored 25.8%. In [24], drone images were created and applied to a modified version of YOLOv5, which scored a mAP of about 50%. Paper [25] compared YOLOv3, YOLOv4, and Faster R-CNN performances during different types of weather (rainy, foggy, snowy). The paper concluded that YOLOv4 outperformed YOLOv3 and Faster R-CNN.

Table 1 shows a summary of recent publications for weather classification and object detection in the AV environment. While standard object detection models primarily focus solely on the detection process, our work and proposed model introduces several key differences compared to recent related studies. First, we have incorporated a new phase in our model called the "Quality Block," designed to assess and enhance the observed scene. Second, we have added an adjustable threshold score to reduce the number of images entering the enhancement phase. Third, our study uniquely addresses sandy weather conditions, which have not been considered in recent publications.

Table 1. Recent publications in the AV environment eliminate sandy weather from their studies. Moreover, there is a gap of combining weather classification and object detection.

Paper	Weather Classification	Object Detection	Drive in Snow	Drive in Rain	Drive in Fog	Drive in Sand
[11]	√	×	√	√	√	×
[13]	√	×	√	√	×	×
[14]	√	×	√	√	√	×
[16]	×	√	√	√	√	×
[18]	√	×	√	×	×	×
[19]	×	√	√	√	√	×
[20]	×	√	√	√	×	×
[21]	×	√	×	×	√	×
[22]	×	√	×	×	√	×
[23]	×	√	×	√	×	×
[24]	×	√	√	√	×	×
[25]	×	√	√	√	√	×
Ours	√	√	√	√	√	√

3. Methodology

Our methodology for developing a model capable of both weather classification and object detection in severe weather started by applying Detection in Adverse Weather Nature (DAWN) dataset [26]. We focused on covering four key weather types (sandy, rainy, foggy, and snowy) with six classes (pedestrian, bicycle, car, motorcycle, bus, and truck). To expand the dataset and introduce a new variation of the existing images, we have included data augmentation in our work. This augmented dataset was combined with the original DAWN dataset to increase the number of training samples. A full description of the augmentation will be provided in Section 6. We then partitioned the combined dataset into training and validation sets. Our split percentage is 80% of the images for training, while (20%) were used for validation and testing (10% for validation and 10% for testing). The training set was used to train both the weather classification and object detection models, while the validation set served the critical role of preventing overfitting. After that, optimization steps are involved to find the best performance of the model by changing the hyperparameters. Lastly, we evaluated the optimized models using the standard mean average precision (mAP), precision, and recall metrics. Figure 4 shows the sequence of our methodology.

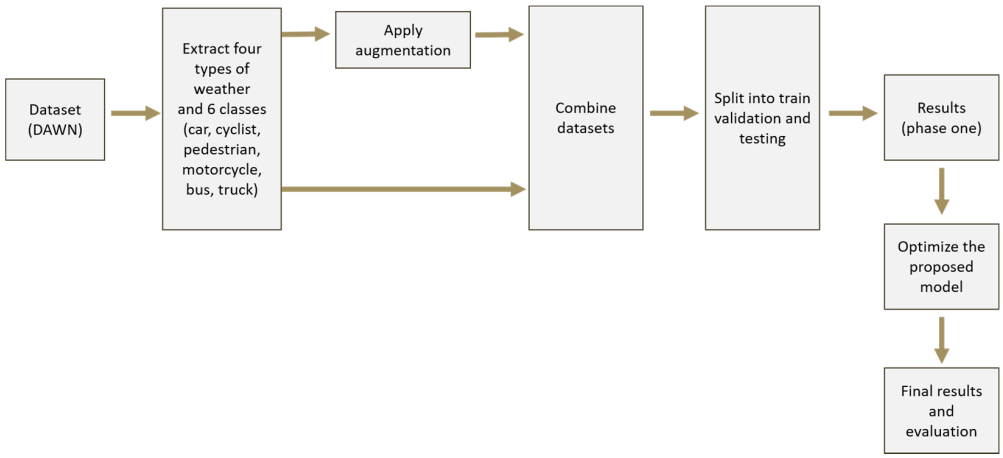


Figure 4. Sequence of our methodology in this paper.

To optimize computational efficiency given limited GPU resources, we employed Google's cloud-based Colab platform as our experimental environment. Colab provides a PyTorch machine learning framework and high-performance GPUs (such as the Tesla T4). Through Colab we were able to effectively execute our experiments, specifically with the integration with CUDA (Compute Unified Device Architecture), which helps in accelerating

the computational process of our pipeline and the CNN part while detecting objects (specifically within tasks like convolution, pooling, normalization, and activation layers).

Various metrics are available for quantifying the efficacy of object detection models. In our paper, we prioritized three main metrics: (a) mean average precision (mAP), (b) precision, and (c) recall. mAP stands as a prevalent evaluation metric within the domain of object detection, offering a holistic assessment of the model's proficiency in object identification and localization. mAP combines precision and recall by computing the average precision (AP) for each object class or category, subsequently deriving the mean across all classes. AP serves as a measure of the detection's quality, encapsulating both the precision of accurately identified objects and the completeness of detection in the scene. Through computation of mAP, our pipeline performance can be numerically compared and evaluated across diverse domains and scenarios.

We also considered precision and recall as indispensable metrics in the context of object detection. Precision is the proportion or the percentage of retrieved elements that are relevant to correct class, while recall measures the percentage of relevant objects that are successfully retrieved. Precision is expressed as the ratio of true positives (TPs) to the sum of true positives and false positives (FPs), represented as:

$$Precision = TP/(TP + FP)$$

Recall is the ratio of TP to the sum of true positives and false negatives (FNs), represented as:

$$Recall = TP/(TP + FP)$$

4. Dataset

For the dataset, as we mentioned earlier, we used DAWN in our development and experimentation. The DAWN dataset covers four types of adverse weather: sandstorm, rain, snow, and fog. Figure 5 shows a sample of the various weather types covered in DAWN. The dataset contains 1027 images covering the four types of weather and different environmental contexts such as highways, freeways, and urban landscapes, ensuring a broad representation of real world scenarios.

Figure 5. The DAWN dataset provides various hard weather conditions such as fog, rain, snow, and sand [26].

Although many other datasets cover adverse weather conditions, the DAWN dataset has the advantage of including sandstorm or sandy weather images, which are often absent

from other datasets. This unique feature of DAWN has allowed our model to address weather classification and object detection in multiple types of geographical environments. The used DAWN dataset, originally consists of 1027 images with a size of 640 × 640.

Image annotation contains the class of the object and the corresponding boundaries of: x, y, width, and height of the bounding box (x_center, y_center, width, height). Figure 6 represents a sample of our labeled images, considered as a ground truth reference.

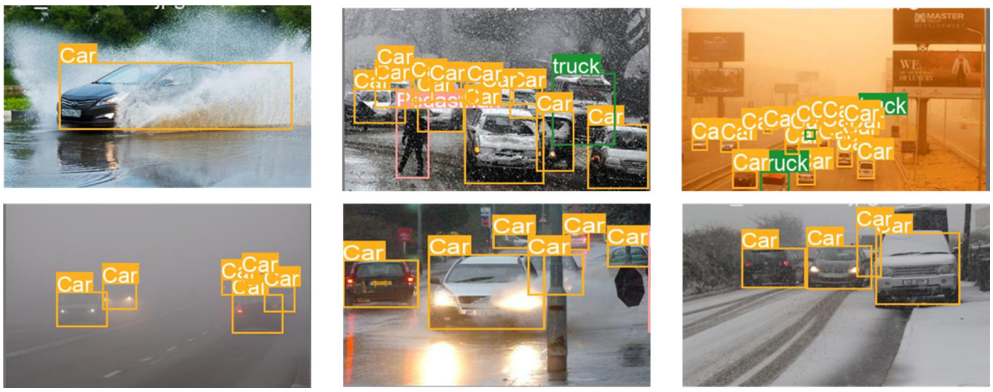


Figure 6. Sample of labeled images.

5. Proposed Model

Our proposed pipeline establishes a comprehensive weather classification and object detection by integrating four main tasks: (1) image quality assessment, (2) image enhancement, (3) weather classification, and (4) object detection. We have combined task 1 and 2 into one block called "Quality Block," while tasks 3 and 4 are combined in another block called "Classify and Detect Block." Once the image enters the model, an assessment will be made to evaluate the image quality. The purpose of this step is to make a decision of whether the captured image is in need of an improvement or not. The Blind Reference-less Image Spatial Quality Evaluator (BRISQUE) method [27] has been employed to quantify image quality. If the entered image has a score that is higher than a threshold point (low quality), the image will be disapproved and transferred to an enhancement stage; otherwise, it will be approved and transferred directly to the classification and detection block. The threshold point can be changed and modified based on the scene situation; for instance, in our experiment we have used a threshold of 42.7, as we will explain in Section 7 "Experiments and Results." It is worth noting that, in the BRISQUE method, generally a lower score indicates better perceptual quality, while a higher BRISQUE score indicates worse perceptual quality. The BRISQUE algorithm has several advantages that made it a suitable solution for our model and for evaluating adverse weather scenes. Firstly, it is a no-reference image quality metric, in that it does not require a perfect reference image for comparison. This is highly advantageous in adverse weather conditions, where obtaining ideal, undistorted images can be challenging, if not impossible. BRISQUE functions by analyzing the natural scene statistics (NSS) of an image and comparing them to the expected statistics of natural (undistorted) images. Any deviations from this naturalness are flagged as indicators of quality degradation—making it a good fit for detecting the kinds of distortions introduced by weather phenomena. Moreover, BRISQUE offers computational efficiency compared to several other options, which might be important when working with large image datasets or in scenarios where real-time quality assessment is desired. For the image enhancement phase, we have used the Super-Resolution Generative Adversarial Network (SRGAN) technique [28], which consists of generator and discriminator networks. The generator network aims to upscale low-resolution images, while the discriminator network aims to refine the generator's output, resulting in improved image clarity. Following the Quality Block, the image is processed by two YOLOv5 networks. One YOLO network, trained extensively on a dataset of weather-labeled images, accurately classifies weather conditions such as sandy, rainy, snowy, or foggy. Simultaneously, a separate

YOLO network, trained to identify and localize objects with bounding boxes, will detect the targeted objects, such as cars, cyclists, pedestrians, motorcycles, buses, and trucks. Generally, our proposed model offers a two-pronged approach, prioritizing image quality before seamlessly transitioning to robust YOLO-based weather classification and object detection for reliable image analysis. Figure 7 shows an illustration of our proposal.

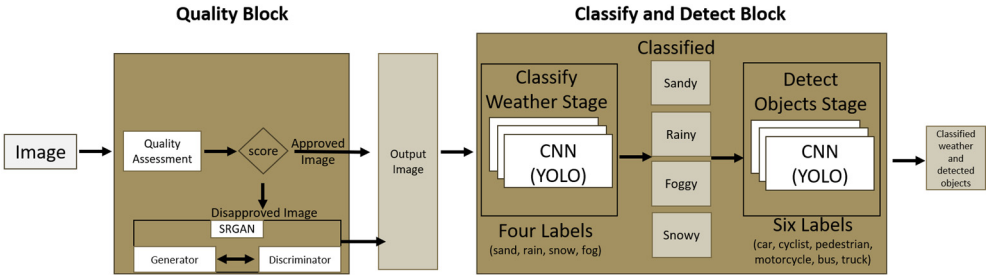


Figure 7. Our proposed model.

6. Augmented DAWN

Given the limited number of adverse weather images in the DAWN dataset, we built a new version of DAWN using augmentation to scale up our experimental dataset. Data augmentation is a widely used method to artificially enlarge datasets by creating new training images from a currently available dataset. Various papers, such as [29,30], and the incremental improvement of YOLOv3 [8] have employed data augmentation either for their weather classification or object detection datasets. Our augmented version has increased the number of DAWN dataset from 1027 images to 2046 images, which is an increase by almost double the current size. Figure 8 shows a general overview of the DAWN dataset before and after our applied augmentation.

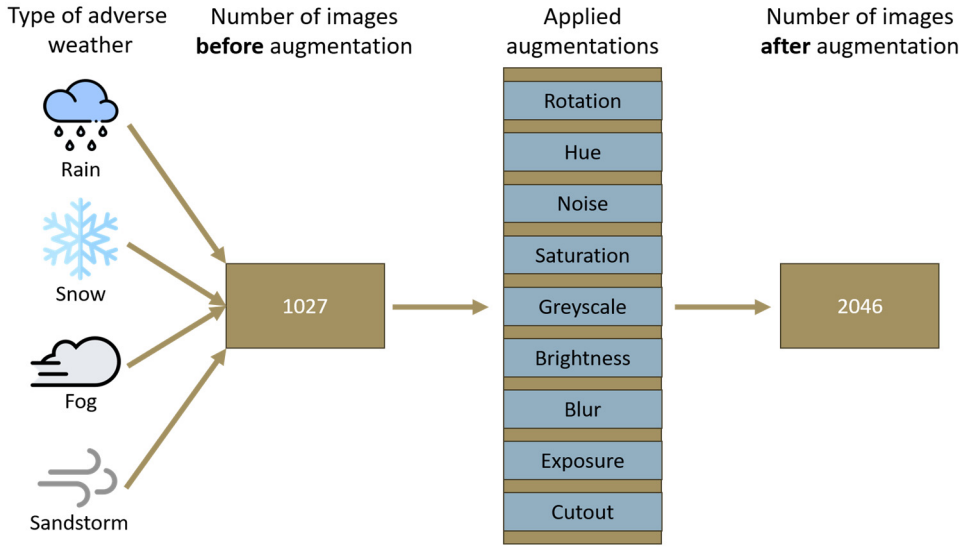


Figure 8. The DAWN dataset increased from 1027 to 2046 images.

Augmentation techniques encompass a range of image adjustments, such as image scaling, rotation, cropping, flipping, color adjustment, noise, or blur, and many other operations. Applying augmenting for weather classification and object detection can be very advantageous for the following reasons:

- Enhancing the diversity and variability of training data, aiding the model's generalization to unrepresented scenarios.
- Boosting the model's resilience against factors affecting object appearance, such as varying lighting conditions, occlusions, or viewpoint changes.

- Addressing class imbalance by oversampling minority classes or undersampling majority ones.
- Mitigating overfitting by introducing regularization and noise into the training data.

The following sections describe the augmentations we performed in this paper.

6.1. Rotation

This is used to present the object from different angles of view. In real-word scenarios, objects might appear at different angles or rotations, and adding this in our augmentation can help the model to better handle these view variations.

6.2. Hue

Hue is a color-based image augmentation technique that alters the hue or color tone of an image while preserving its brightness and saturation.

6.3. Noise

We have also incorporated synthetic noise into our augmentation process to expand our dataset. This type of augmentation enhances our model's resilience to noise and enhances its capability to adapt to new data or scenarios.

6.4. Saturation

Saturation adjusts the intensity of colors within an image. By saturating an image, we effectively scale the pixel values by a random factor within a specified range. Increasing the saturation value of an image can make the colors more vibrant and vivid, while decreasing it can make the colors more subdued and muted. We augmented the saturation of our dataset by approximately 25%.

6.5. Grayscale

We incorporated grayscale augmentation, which converts an image into grayscale. This technique is commonly used to increase the contrast of an image and enhance its details.

6.6. Brightness

By randomly increasing the brightness of images, we subjected our model to a broader range of lighting conditions, thereby enhancing its resilience to changes in illumination. We augmented the images, rendering them approximately 15% brighter.

6.7. Blur

Blur is used to introduce out-of-focus effects into images. For our augmented data, we used Gaussian blur with up to 1.25 px.

6.8. Exposure

Additionally, we artificially modified the exposure level of the images, setting it in the range of 10% to −10%.

6.9. Cutout

We have also cut small parts of objects in the scene. The purpose of this is to add occlusion to our experiment, which is to block, cover, or obscure an object from the camera view.

Table 2 shows our augmentation setting values and their impacts on images.

Table 2. Summary of applied augmentations and their impact on image.

Augmentation	Value	Impact
Rotation	90 degree	Helps the model to be insensitive to camera orientation
Hue	15%	Random adjustment of colors
Noise	Random noise	More obstacles added to image

Table 2. *Cont.*

Augmentation	Value	Impact
Saturation	25%	Changes the intensity of the pixels
Grayscale	15%	Converts image to single channel
Brightness	15%	Image appears lighter
Blur	1.25px	Averages pixel values with neighboring ones
Exposure	10%	Resilient to lighting and camera setting changes
Cutout	Cut random parts of the image	More resilient to detect half objects

7. Experiments and Results

To test our model, we have conducted several experiments, starting by setting set our BRISQUE threshold score to 42.75. This score is the average quality score for the DAWN dataset, and any image above this average score will go through the enhancement stage. Table 3 explains the reason behind choosing 42.75 as our threshold point and illustrates the impact of image quality by measuring BRISQUE scores before and after augmentation. The table compares images from the DAWN dataset (sandstorms, rain, snow, and fog) with our extended augmented images of the same dataset aiming to simulate the adverse weather conditions. Across all weather conditions, augmented images generally exhibit higher BRISQUE scores, indicating a decline in image quality compared to the original DAWN images. As the table shows, augmented images are worse by about 9% when it comes to the average scene quality. This low quality of the augmented images can be attributed to the performed augmentations (blur, hue, saturation, noise, cut, brightness, and exposure), which are usual effects during adverse weather. The observed differences underscore the importance of designing a quality assessment model to preserve image quality, particularly in adverse weather conditions where visual clarity is crucial for accurate scene observation and object detection.

Table 3. Comparison of image quality with and without augmentation. The results shows an average image quality of 46.59 for the augmented DAWN dataset compared to 42.75 for the original DAWN dataset.

	Sandy	Foggy	Snowy	Rainy	Average
DAWN images Quality Score	44.05	45.21	40.18	41.57	42.75
Augmented DAWN images Quality Score	48.71	49.83	43.19	44.64	46.59

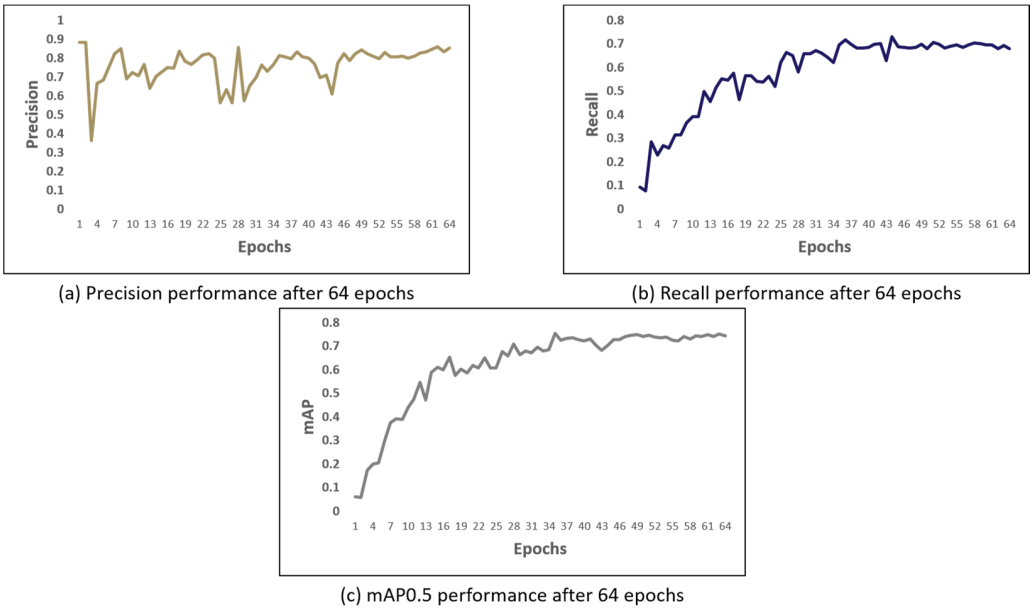
The experimental scenario for the augmented DAWN dataset was executed within the Google Colab environment, harnessing the computational power of a Tesla T4 GPU. We have made several modification to the YOLOv5 architecture, aiming to create a suitable model for our domain. This modification includes changing the activation functions and test the model with SiLU and LeakyRelu functions. We also have modified the backbone and head to test the performance of BottleneckCSP and C3 architectures. In addition to that, hyperparameters such as epochs and batch size have been changed throughout our experiments.

After designing our model, we initiated our experimental phase by implementing BottleneckCSP as both the backbone and head architecture. Our model demonstrated promising results, achieving a mean average precision (mAP) of 55.6% and 45.6% when trained for 128 epochs with a batch size of 32, utilizing SiLU and LeakyReLU activation functions, respectively. From Table 4, it can be clearly seen that, when we implemented BottleneckCSP in our model, the mAP was increasing for SiLU and LeakyRelu functions whenever we increased the number of epochs. It can be also seen that LeakyRelu has lower performance than SiLU with the BottleneckCSP backbone and head. Table 4 shows the complete results of our model using BottleneckCSP.

Table 4. Performance of our model using BottleneckCSP as a backbone and head.

Backbone and Head	Activation Function	Epoch	Batch	mAP
BottleneckCSP	SiLU	32	16	33.7%
BottleneckCSP	SiLU	32	32	34.5%
BottleneckCSP	SiLU	64	16	40.2%
BottleneckCSP	SiLU	64	32	43.9%
BottleneckCSP	SiLU	128	16	55.2%
BottleneckCSP	SiLU	128	32	55.6%
BottleneckCSP	LeakyRelu	32	16	24.0%
BottleneckCSP	LeakyRelu	32	32	25.1%
BottleneckCSP	LeakyRelu	64	16	34.7%
BottleneckCSP	LeakyRelu	64	32	34.9%
BottleneckCSP	LeakyRelu	128	16	38.7%
BottleneckCSP	LeakyRelu	128	32	45.6%

We continued our experiments by moving to include Concentrated-Comprehensive Convolution (C3) [31] as a backbone and head in our proposed model. The model achieved a better result, achieving 71.8% mAP using SiLU with only 32 epochs and 16 batches, as Table 5 shows. This score is higher than LeakyRelu by 7.4 percentage points, with the same metrics. This result is also higher than the highest score achieved using the BottleneckCSP backbone (Table 4), which was 55.6%. We continued to increase the number of epochs and batches until we achieved 74.6% after 64 epochs with 16 batches, which is the highest score mAP in this paper. As we will discuss below, this score is the highest mAP score compared with other and recent object detection publications that used DAWN as a base dataset. The mAP, precision, and recall of our model can be seen in Figure 9. The top left chart shows the precision result, with a score of 85%, while the top right chart shows the recall result reaching 68%. The bottom chart shows the resulting mAP, which reached 74.6%. Figure 10a shows our F1 score, and we can clearly see that the peak for most classes occurs at confidence thresholds between 0.4 and 0.6. This suggests that setting the model's confidence threshold within this range would likely yield the best balance between precision and recall. It also shows that the model excels at detecting cars, while it struggles more with trucks, with overall performance following a similar pattern to the average across all classes. Figure 10b represents our mAP, with intersection over union thresholds from 0.5 to 0.95.



(a) Precision performance after 64 epochs

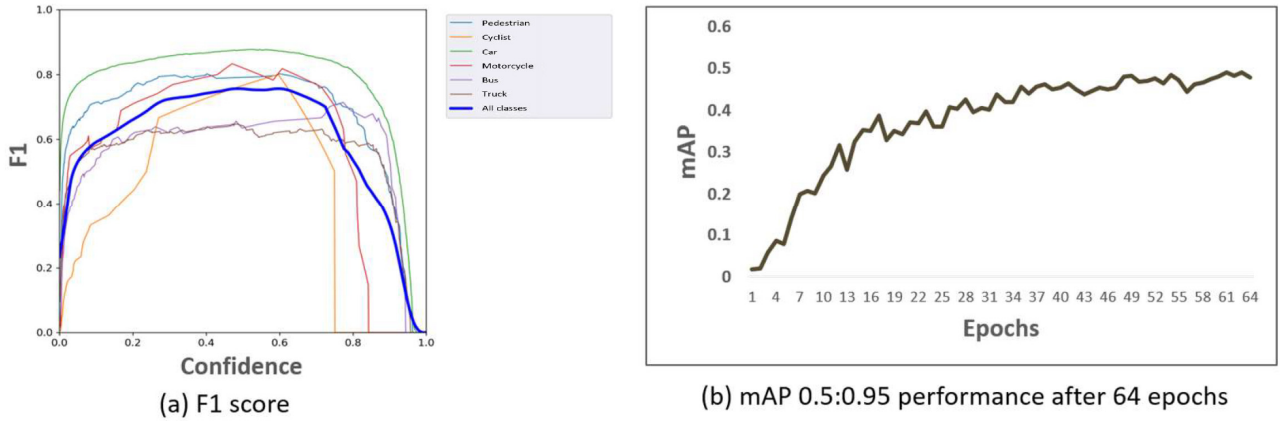
(b) Recall performance after 64 epochs

(c) mAP0.5 performance after 64 epochs

Figure 9. Precision, recall, and mAP at 0.5 of our model after 64 epochs.

Table 5. Performance of our model using C3 as a backbone and head.

Backbone and Head	Activation Function	Epoch	Batch	mAP
C3	SiLU	32	16	71.8%
C3	SiLU	32	32	68.6%
C3	SiLU	64	16	74.6%
C3	SiLU	64	32	74.1%
C3	SiLU	128	16	74.0%
C3	SiLU	128	32	73.1%
C3	LeakyRelu	32	16	64.4%
C3	LeakyRelu	32	32	67.1%
C3	LeakyRelu	64	16	62.9%
C3	LeakyRelu	64	32	63.2%
C3	LeakyRelu	128	16	72.9%
C3	LeakyRelu	128	32	72.4%



(a) F1 score (b) mAP 0.5:0.95 performance after 64 epochs

Figure 10. F1 score and mAP at 0.5:0.95 of our model after 64 epochs.

The previously mentioned tables, Tables 4 and 5, highlight a notable observation: increasing the number of epochs does not consistently correlate with higher mean average precision (mAP). Surprisingly, our model achieved its highest mAP score when trained for 64 epochs rather than 128 epochs, contrary to the initial expectation. This finding highlights the importance of careful hyperparameter tuning and validation experimentation in identifying the most effective training regimen for a given model architecture and dataset. Among recent publications utilizing the DAWN dataset, our method's mean average precision (mAP) of 74.6% is the highest achieved, as detailed in Table 6.

Table 6. Comparison of our result with some recent publications that used the DAWN dataset.

Ref.	mAP	Dataset	Aim
[32]	32.75%	DAWN	Ensemble approach to improve object detection in AVs under adverse weather conditions.
[33]	Fog 29.66%	DAWN	Modifying YOLO and using several datasets to detect objects in the AV environment.
	Rain 41.21%		
	Snow 43.01%		
	Sand 24.13%		
[34]	39.19%	DAWN	Architecture for constructing datasets using GAN and CycleGAN.
[35]	55.85%	DAWN	Low-light Detection Transformer (LDETR) to improve detection performance.
[36]	72.8%	DAWN	Improving YOLO using metaheuristic algorithms.
Ours	74.6%	DAWN	Modifying YOLO and using DAWN dataset to classify weather and detect objects in the AV environment.

For weather classification evaluation, the proposed model scored an accuracy of 74.3% after 64 epochs, as Figure 11 shows. The model has successfully classified most of the scenes; however, there are some cases where the model failed to classify the true weather. For instance, if we look at Table 7, which shows the experimental result of weather classification, in image number 5 the true weather was a strong sandstorm, whereas the model classified it as foggy weather. This case is an example of where the brightness and lighting of the scene could be challenging for weather classification models in adverse weather.

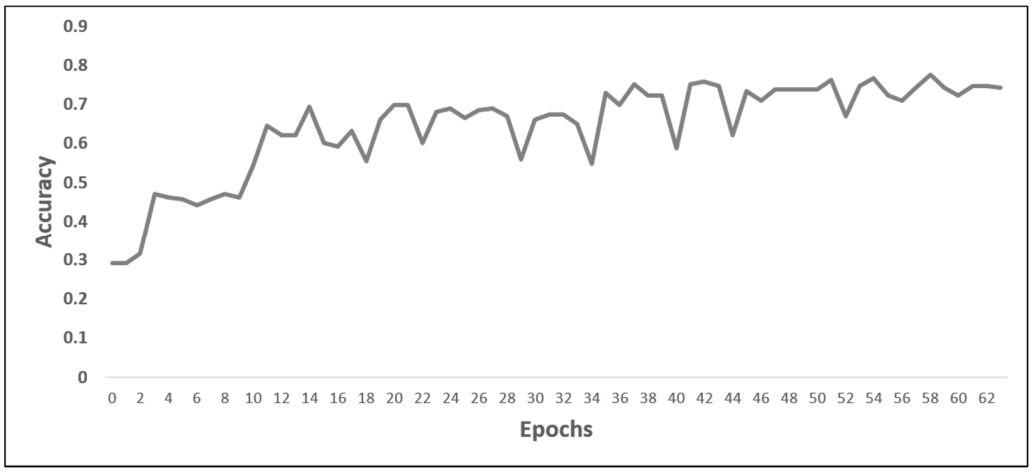


Figure 11. For weather classification, our model scored 74.3% after 64 epochs.

Table 7. Classification results.

Number	Image	Ground	Classified
1		Sandy weather	Sandy weather
2		Foggy weather	Foggy weather
3		Rainy weather	Rainy weather

Table 7. *Cont.*

Number	Image	Ground	Classified
4		Snowy weather	Snowy weather
5		Sandy weather	Foggy weather

Figures 12–14 are examples from our experiments where the model successfully classified the type of weather in the scene and detected the objects. The top left corner shows the probability scores for the current weather, ranking them from highest probability to lowest. For instance, in Figure 12, the model classified the scene as sandy weather by 87%, which is correct and matching the ground truth. It is also successfully detected the two vehicles appearing in the scene, with detection percentages of 96% and 94%, respectively.

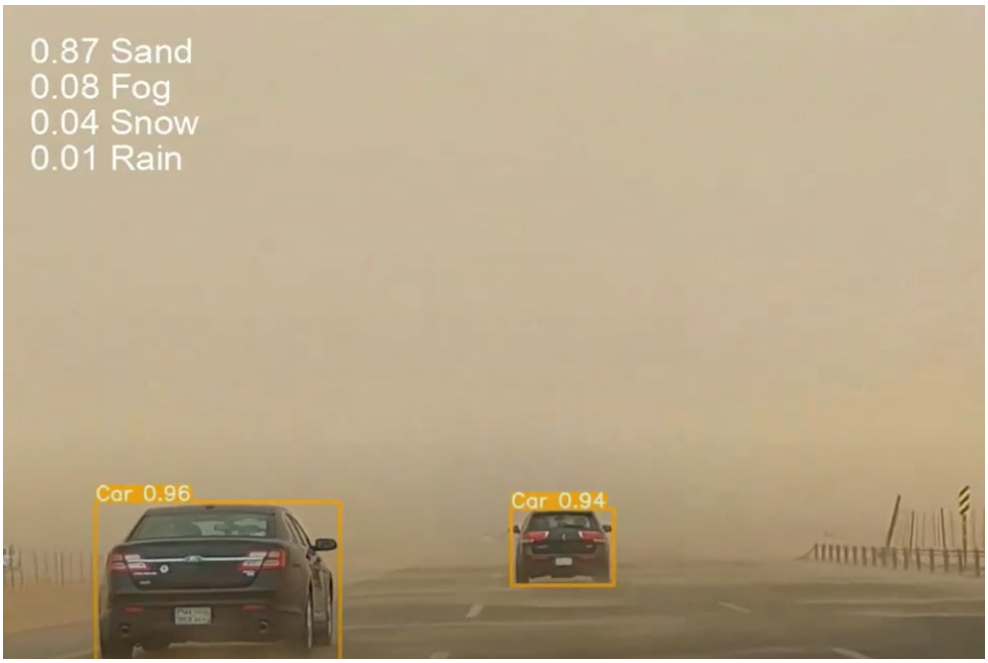


Figure 12. The model has successfully classified sandy weather and detected objects in the scene.

Figure 13. The model has successfully classified foggy weather and detected objects in the scene.

Figure 14. The model has successfully classified rainy weather and detected objects in the scene.

8. Discussion

While the preceding section demonstrates the potential of our method to detect various vehicles in adverse weather, in the following points we will discuss key insights and observations that emerged during the development of this work:

- If we look at our F1 score (Figure 10), the "car" class consistently achieves the highest F1 scores across different confidence levels, indicating that the model is particularly adept at detecting cars accurately. Conversely, the "truck" class generally exhibits the lowest F1 scores, suggesting that the model might have more difficulty distinguishing trucks, or faces more false positives/negatives in this category. The "all classes" curve represents the average performance across all object classes, demonstrating a similar trend to the individual classes, with the peak F1 score around the 0.6 confidence threshold.
- As Table 6 shows, our proposed work achieved a mAP of 74.6%. This result surpasses the performance of other publications on the DAWN dataset, including ensemble methods [32], YOLO modifications [33], GAN-based architectures [34], the LDETR transformer [35], and YOLO enhanced with metaheuristic algorithms [36].
- Notably, DAWN is a very challenging dataset, as corroborated by our own experience and underscored by the observations of the authors in [33], who remarked: "[w]e find the DAWN dataset a bit more challenging than the others." This challenge is due to the fact that some images and objects are characterized by exceedingly low visibility, which is a factor that may impact the resulting score of any developed model.
- The domain of object detection in adverse weather still requires more reliable datasets that provide sufficient variability to cover various object appearances, lighting conditions, and occlusions. Creating such datasets is time-consuming and costly. A recently published paper by Liu et al. [37] demonstrated a simulator-based approach that allows easy manipulation of environmental conditions, object placement, and camera perspectives. Using simulator-based data collection opens the door to diverse and comprehensive datasets without extensive real-world data gathering. This approach can expedite data collection by setting up and executing various adverse weather scenarios without, for instance, waiting for the weather's seasonal changes. Furthermore, it offers data scalability, overcoming the geographical constraints of real-world data collection.
- While existing recent publications and public datasets offer valuable resources for object detection in various weather conditions, there is a clear need for more work that includes sandy weather scenarios.
- Combining images with LiDAR using fusion can be a promising approach for enhancing object detection in autonomous vehicle environments. Recent studies, such as those by Dai et al. [38] and Liu et al. [39], have demonstrated that this technique significantly improves object detection in challenging environments by leveraging the complementary features of both LiDAR and cameras. Cameras provide a cost-effective, lightweight solution that captures rich color and texture details, aiding in the classification and identification of objects. On the other hand, LiDAR offers precise distance measurements and 3D spatial information, which are particularly useful in low-visibility conditions where cameras may struggle. By fusing the data from both sensors, the accuracy and robustness of object detection systems can be greatly enhanced.
- We extended our experiments to test our model using the UAVDT dataset [40]. The original UAVDT dataset comprises over 77,000 images captured in daylight, night, and foggy weather conditions. After running the experiment for 64 epochs, we achieved the following results: mAP of 94.1%, recall of 90.8%, and precision of 97.0%. We believe that the UAVDT dataset requires additional preprocessing before it can be fully utilized. For instance, adjusting the time frame for capturing images could help diversify the resulting images.
- Synthetic data can be used to address the challenges and limitations of real-word datasets. In a recent publication [41], the authors proposed CrowdSim2, a synthetic dataset, for object detection tasks, particularly detection of people and vehicles. Such a technique can be very beneficial for the AV domain by offering a controlled environment where factors like weather conditions, object density, and lighting can be

precisely manipulated, enabling the testing of object detection models under various scenarios. Additionally, it can be used to simulate rare but critical events, such as accidents or unusual obstacles, which may be underrepresented in real-world datasets.

9. Conclusions

Classifying weather and detecting objects in severe weather environments is a critical and challenging task. In this paper we introduced a multi objectives model that integrates weather classification and object detection and treats them as a unified problem within the domain of autonomous vehicle perception systems. Our model consists of two main blocks. First, the Quality Block checks image quality based on BRISQUE score, and if the image has a score that is higher than the threshold then it proceeds further to be enhanced by an SRGAN method. Second, the Classify and Detect Block classifies four types of adverse weather (snowy, rainy, foggy, and sandy) and detects six classes (car, cyclist, pedestrian, motorcycle, bus, and truck). During our development, we utilized the challenging DAWN dataset as our source of images and employed YOLO as the base structure for classification and detection. The experimental results show that our model achieved a mean average precision (mAP) of 74.6% for detecting objects using the YOLO architecture with C3 architecture as a backbone and SiLU as an activation function. Additionally, for classifying the weather of the scene, our model attained an accuracy of 74.3%, which closely aligns with the mAP. Having said that, there are still some challenges in the domain that should be considered while developing detection and classification models. Changes of the scene characteristics such as lighting and cloudiness lead to wrong classification of the correct weather.

10. Future Work

Adverse weather is still a very challenging domain in AV environments. To achieve the highest level of automation, camera sensors are in need of a robust system that is capable of navigating safely in diverse weather scenarios and capable of accurately observing the surroundings. In the future, we will extend our domain to include additional datasets that could be merged with the current DAWN dataset. This could lead us to expand our detection classes to include more detailed and new classes that we observe in real driving environment, such as traffic lights, children, domestic animals (like dogs), and law enforcement personnel (such as police officers). Each of these classes represents integral components of the road scene, and accurately detecting and responding to their presence is essential for ensuring the safety and efficiency of autonomous driving systems. By incorporating these additional classes into our detection framework, we aim to enhance the overall mAP. Additionally, we aim to enhance the perceptual capabilities of autonomous systems through perceptual fusion, which involves combining information from multiple sensors, such as cameras, LiDAR, radar, and ultrasonic sensors, to create a comprehensive and accurate representation of the surrounding environment. By developing such a robust system, we believe that we can mitigate the impact of adverse weather conditions on sensor performance and enhance the reliability and robustness of general AV perception systems.

Author Contributions: Software N.A.; data analysis, N.A., A.A. and A.B.; data curation, N.A.; methodology, N.A; writing—original draft preparation, N.A.; writing—review and editing, A.A. and A.B.; supervision, A.A. and A.B.; conceptualization, N.A., A.A. and A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

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

References

1. *J3016_202104*; Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. Society of Automotive Engineers (SAE): Warrendale, PA, USA, 2021.
2. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
3. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Seville, Spain, 17–19 March 2015; pp. 1440–1448.
4. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
5. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Jaipur, India, 18–21 December 2017; pp. 2961–2969.
6. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Penang, Malaysia, 5–8 November 2017; pp. 2117–2125.
7. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Bangalore, India, 20–21 May 2016; pp. 779–788.
8. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
9. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
10. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
11. Dhananjaya, M.M.; Kumar, V.R.; Yogamani, S. Weather and light level classification for autonomous driving: Dataset, baseline and active learning. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 2816–2821.
12. Kondapally, M.; Kumar, K.N.; Vishnu, C.; Mohan, C.K. Towards a Transitional Weather Scene Recognition Approach for Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2023**, *25*, 5201–5210.
13. Ibrahim, M.R.; Haworth, J.; Cheng, T. WeatherNet: Recognising weather and visual conditions from street-level images using deep residual learning. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 549. [CrossRef]
14. Xia, J.; Xuan, D.; Tan, L.; Xing, L. ResNet15: Weather recognition on traffic road with deep convolutional neural network. *Adv. Meteorol.* **2020**, *2020*, 6972826. [CrossRef]
15. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Bangalore, India, 20–21 May 2016; pp. 770–778.
16. Cao, Y.; Li, C.; Peng, Y.; Ru, H. MCS-YOLO: A multiscale object detection method for autonomous driving road environment recognition. *IEEE Access* **2023**, *11*, 22342–22354.
17. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
18. Pan, G.; Fu, L.; Yu, R.; Muresan, M.I. Winter Road Surface Condition Recognition Using a Pre-Trained Deep Convolutional Neural Network. In Proceedings of the Transportation Research Board 97th Annual Meeting, Washington, DC, USA, 7–11 January 2018; pp. 838–855.
19. Wang, R.; Zhao, H.; Xu, Z.; Ding, Y.; Li, G.; Zhang, Y.; Li, H. Real-time vehicle target detection in inclement weather conditions based on YOLOv4. *Front. Neurorobotics* **2023**, *17*, 34.
20. Li, X.; Wu, J. Extracting High-Precision Vehicle Motion Data from Unmanned Aerial Vehicle Video Captured under Various Weather Conditions. *Remote Sens.* **2022**, *14*, 5513. [CrossRef]
21. Liu, W.; Ren, G.; Yu, R.; Guo, S.; Zhu, J.; Zhang, L. Image-adaptive YOLO for object detection in adverse weather conditions. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, 27 February–2 March 2022; Volume 36, pp. 1792–1800.
22. Huang, S.C.; Le, T.H.; Jaw, D.W. DSNet: Joint semantic learning for object detection in inclement weather conditions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 2623–2633.
23. Sharma, T.; Debaque, B.; Duclos, N.; Chehri, A.; Kinder, B.; Fortier, P. Deep learning-based object detection and scene perception under bad weather conditions. *Electronics* **2022**, *11*, 563. [CrossRef]
24. Jung, H.K.; Choi, G.S. Improved yolov5: Efficient object detection using drone images under various conditions. *Appl. Sci.* **2022**, *12*, 7255. [CrossRef]
25. ABDULGHANI, A.M.A.; DALVEREN, G.G.M. Moving Object Detection in Video with Algorithms YOLO and Faster R-CNN in Different Conditions. *Avrupa Bilim Ve Teknoloji Dergisi* **2022**, *33*, 40–54. [CrossRef]
26. Kenk, M.A.; Hassaballah, M. DAWN: Vehicle detection in adverse weather nature dataset. *arXiv* **2020**, arXiv:2008.05402.
27. Mittal, A.; Moorthy, A.K.; Bovik, A.C. No-reference image quality assessment in the spatial domain. *IEEE Trans. Image Process.* **2012**, *21*, 4695–4708. [CrossRef] [PubMed]

28. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Penang, Malaysia, 5–8 November 2017; pp. 4681–4690.
29. Zoph, B.; Cubuk, E.D.; Ghiasi, G.; Lin, T.Y.; Shlens, J.; Le, Q.V. Learning data augmentation strategies for object detection. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 566–583, Part XXVII 16.
30. Volk, G.; Müller, S.; Von Bernuth, A.; Hospach, D.; Bringmann, O. Towards robust CNN-based object detection through augmentation with synthetic rain variations. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC); IEEE: Piscataway, NJ, USA, 2019; pp. 285–292.
31. Park, H.; Yoo, Y.; Seo, G.; Han, D.; Yun, S.; Kwak, N. C3: Concentrated-comprehensive convolution and its application to semantic segmentation. *arXiv* **2018**, arXiv:1812.04920.
32. Walambe, R.; Marathe, A.; Kotecha, K.; Ghinea, G. Lightweight object detection ensemble framework for autonomous vehicles in challenging weather conditions. *Comput. Intell. Neurosci.* **2021**, *2021*, 5278820. [CrossRef] [PubMed]
33. Farid, A.; Hussain, F.; Khan, K.; Shahzad, M.; Khan, U.; Mahmood, Z. A fast and accurate real-time vehicle detection method using deep learning for unconstrained environments. *Appl. Sci.* **2023**, *13*, 3059. [CrossRef]
34. Musat, V.; Fursa, I.; Newman, P.; Cuzzolin, F.; Bradley, A. Multi-weather city: Adverse weather stacking for autonomous driving. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2906–2915.
35. Tiwari, A.K.; Pattanaik, M.; Sharma, G. Low-light DEtection TRansformer (LDETR): Object detection in low-light and adverse weather conditions. *Multimed. Tools Appl.* **2024**, 1–18. [CrossRef]
36. Özcan, I.; Altun, Y.; Parlak, C. Improving YOLO Detection Performance of Autonomous Vehicles in Adverse Weather Conditions Using Metaheuristic Algorithms. *Appl. Sci.* **2024**, *14*, 5841. [CrossRef]
37. Liu, S.; Zhang, H.; Qi, Y.; Wang, P.; Zhang, Y.; Wu, Q. Aerialvln: Vision-and-language navigation for uavs. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 15384–15394.
38. Dai, Z.; Guan, Z.; Chen, Q.; Xu, Y.; Sun, F. Enhanced Object Detection in Autonomous Vehicles through LiDAR—Camera Sensor Fusion. *World Electr. Veh. J.* **2024**, *15*, 297. [CrossRef]
39. Liu, H.; Wu, C.; Wang, H. Real time object detection using LiDAR and camera fusion for autonomous driving. *Sci. Rep.* **2023**, *13*, 8056. [CrossRef] [PubMed]
40. Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; Tian, Q. The unmanned aerial vehicle benchmark: Object detection and tracking. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 370–386.
41. Foszner, P.; Szczesna, A.; Ciampi, L.; Messina, N.; Cygan, A.; Bizon', B.; Cogiel, M.; Golba, D.; Macioszek, E.; Staniszewski, M. CrowdSim2: An open synthetic benchmark for object detectors. *arXiv* **2023**, arXiv:2304.05090.

electronics

MDPI

Article

Impact of Communication Link Overload on Power Flow and Data Transmission in Cyber–Physical Power Systems

Xinyu Liu [1,2,3], Yan Li [1,2,3,*] and Tianqi Xu [1,2,3]

[1] School of Electrical and Information Technology, Yunnan Minzu University, Kunming 650504, China; xyliu7918@gmail.com (X.L.); xu.tianqi@ymu.edu.cn (T.X.)
[2] Yunnan Key Laboratory of Unmanned Autonomous System, Kunming 650504, China
[3] Key Laboratory of Cyber–Physical Power System of Yunnan Colleges and Universities, Kunming 650504, China
* Correspondence: yan.li@ymu.edu.cn

Abstract: The volume of flow demand in cyber-physical power systems (CPPSs) fluctuates unevenly across coupled networks and is susceptible to congestion or overload due to consumers' energy demand or extreme disasters. Therefore, considering the elasticity of real networks, communication links with excessive information flow do not immediately disconnect but have a certain degree of redundancy. This paper proposes a dynamic cascading failure iterating model based on the distribution of information flow overload in a communication network and power flow betweenness in the physical power grid. First, a nonlinear load capacity model of a communication network with overload and weighted edges is introduced, fully considering the three link states: normal, failure, and overload. Then, flow betweenness substitutes for branch flows in the physical power network, and power flow on failed lines is redistributed using the load capacity model, simplifying the calculations. Third, under the influence of coupling relations, a comprehensive model based on improved percolation theory is constructed, with attack strategies formulated to more accurately assess the coupled networks. Simulations on the IEEE-39 bus system demonstrate that considering the overload capacity of communication links on a small scale enhances the robustness of coupled networks. Furthermore, deliberate link attacks cause more rapid and extensive damage compared to random attacks.

Keywords: overloaded edges; information flow; power flow; cascading failures; improved percolation theory; cyber–physical power systems

Citation: Liu, X.; Li, Y.; Xu, T. Impact of Communication Link Overload on Power Flow and Data Transmission in Cyber–Physical Power Systems. *Electronics* **2024**, *13*, 3065. https://doi.org/10.3390/electronics13153065

Academic Editor: Christos J. Bouras

Received: 8 July 2024
Revised: 30 July 2024
Accepted: 31 July 2024
Published: 2 August 2024

1. Introduction

1.1. Background

With the development of the smart grid and the energy internet, the power system has become deeply coupled with the information system. The power system on the physical side and the communication system on the information side have gradually evolved into the cyber–physical power system (CPPS) [1]. While the coupled system has brought numerous benefits, it has also increased the risk of cascading failures across space. Vulnerabilities in the two systems through an overlapping network will increase the risk of fault propagation, such that even a single edge or node failure can impact the entire network, often leading to a global collapse [2,3]. For instance, the massive blackout in the western United States in 2003, the blackout in Ukraine in 2015, and the 815 blackout in Brazil in 2023 [4,5] were all caused by the failure of specific edges in the information network. These failures propagated to the power grid through functional coupling, ultimately resulting in the simultaneous paralysis of both systems.

1.2. Related Works

The analysis of past cases shows that when the communication system fails or is attacked, data packets can be lost or manipulated, preventing closed-loop control. Due to the cyber–physical coupling connection, the failure will propagate to affect power nodes in the physical network with a certain probability. Then, the failure continues spreading in the physical network, ultimately causing serious damage to the system [6–8]. Therefore, modeling the physical power grid, communication network, and its coupling connection is essential for understanding the propagation process of cascading failure across space.

In [9], a one-to-one coupling model between power and communication nodes was proposed, analyzing the robustness of the cascading failure system after the removal of a small fraction of nodes based on its topological model. In [10], the authors studied the robustness of a dual-layer scale-free communication network based on percolation theory. Based on [10], Ref. [11] considered the interactions among nodes in different layers as heterogeneous, studying a type of cascading dynamics in dual-layer networks that exhibit both interdependence and connectivity. Based on [9,10], Chen et al. [12] differentiated the nodes in the physical power grid into generator and load nodes, proposing a new interactive mechanism for cascading failures. In [13], the operational characteristics and topological structure of the transmission network were integrated to establish a cascade failure model for random faults in transmission lines under different coupling strategies, aiming to obtain an optimal robust coupled network. However, the coupling models established in these studies focus solely on topological structures, neglecting the operational characteristics of both sides of the coupled networks. In [14,15], power flow optimization in the physical power network was considered and the results of vulnerability under different strategies and information network topologies were compared, but the operational characteristics of the information network were not considered. In [16], the dynamic propagation of cascade failures between the power grid and the communication network was studied, considering the characteristics of power flow and data flow in two different systems, but the impact of data overload in the communication network on the coupled network was not considered. In [17], the recovery characteristics of different coupling strengths and network topologies based on a load-related cascade model were studied. Although the overload state of nodes was considered in this model, the extra load was not redistributed. Based on [16,17], Ding et al. [18] proposed an improved cascading failures model. This model considers the overload state and recovery process of cyber nodes, as well as the optimization of power flow in the physical layer and the redistribution of information flow during fault propagation. Building on this, Wang et al. [19] employed an AC power flow model to characterize the operational characteristics of the power grid, enhancing the accuracy of the power grid model. Simultaneously, it constructed a weighted communication network with control centers and applied a flow redistribution model. In [20], the authors proposed two types of strong and weak dependency models and analyzed the robustness changes of the coupling network using a congestion-aware load balancing scheme under initial random faults in the power layer. However, data flows in the communication layer were not considered. The authors of [21,22] considered communication node failures and established an improved cascade failure model based on the physical layer load distribution. In [23], the authors' proposed model considered the practical differences between a communication network and a power network in terms of network structure, physical operation, and dynamic behavior, focusing on analyzing faults occurring on the power grid side. From these studies, it is evident that most scholars have paid less attention to transmission delays caused by traffic overloads in communication networks and the establishment of coupled models that incorporate the operational characteristics of both networks.

1.3. Motivation

In fact, many connected edges often possess redundant capacity. For example, Figure 1a illustrates a communication network with five nodes. The matrix F represents the information flow transmission demand matrix, where elements F_{ij} indicate the information flow

demand that needs to be transmitted from the source to the destination. Each link e_{ij} is associated with its quality attribute q_{ij} [24,25], where the operational level of a link e_{ij} is defined as the ratio of link capacity to link load. Assuming the failure threshold of the link is ρ, if the initial faults in the communication network are links with $\rho < 0.5$, these links will be removed from the original network (e.g., remove 1→2, 2→3). At this point, the information flow transmission on the communication link is not affected (links 1→4→2 and 2→1→3 still exist), but links 1→4 and 2→1 are in an overloaded state. The information flow demand on the original link 1→2 will be redistributed to the link 1→4→2. Although the network is not immediately affected and the overloaded links do not fail, the transmission quality will continue to decrease. When increasing the threshold to $\rho = 0.7$, the operating efficiency of link 1→4 drops below the critical threshold, causing the edge's state to change from overloaded to failed, and it is removed from the original network. At this point, 25 information flow transmission demands are affected (highlighted in red in the figure). When the threshold is further increased to $\rho = 0.9$, links with $q_{ij} < 0.9$ are removed. As shown in Figure 1d, only eleven units of traffic demand can be effectively transmitted to the control center. It can be seen that during the process of changing the threshold and removing links, if the overloaded state and the transmission flow demand of the links are not considered, the network will collapse prematurely, leading to significant losses in the power grid.

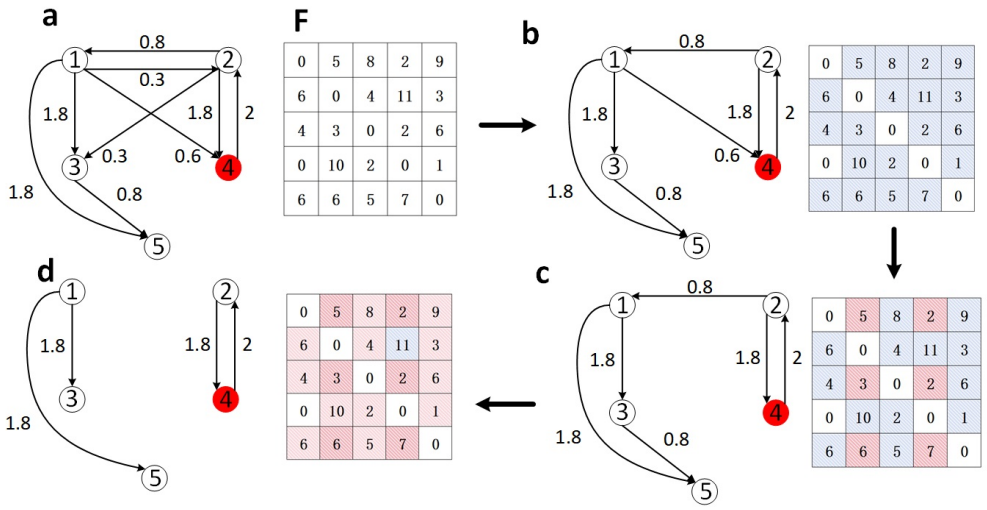


Figure 1. Power communication network diagram of link failure based on information flow demand. (**a**) Communication network G with size $n = 5$, where node four is the control center and the others are regular transmission nodes. The quality q_{ij} of each line e_{ij} is marked beside the link. The matrix F represents the information flow transmission demand. (**b**) Assuming the initial faults in the communication network are links with $\rho < 0.5$. (**c**) Increasing the threshold to $\rho = 0.7$, while removing links with $q_{ij} < 0.7$. (**d**) When the threshold is further increased to $\rho = 0.9$, links with $q_{ij} < 0.9$ are removed and only eleven units of traffic demand can be effectively transmitted to the control center.

Typically, when an N-1 failure occurs in the power grid, flow convergence must be recomputed for each scenario. If the flow converges, it must be determined whether the flow on each line exceeds its limits; if so, the affected line should be cut. If it does not converge, load shedding or generator output adjustments are typically implemented to balance the power flows. Enumerating all scenarios can be both complex and time-consuming. Therefore, this paper proposes applying line flow betweenness [26] to the load–capacity model of the power system, utilizing it as the power load on the line. This approach effectively reflects and quantifies the role of lines in transmitting power from generators to loads, also considering the impact of the maximum available transmission power between generators and loads on the critical lines. This physical background aligns more closely with

the actual operation of power systems and better reflects their operational characteristics. When modeling the information flow transmission process, distinguishing it from the power network is crucial. Information layer links will not disconnect due to information flow overload but will cause congestion if the flow is excessive. It is necessary to account for the overloaded state when redistributing information flow from non-operational links to neighboring links to develop a more realistic power–communication network flow model. Therefore, this paper, considering link interruptions in the communication system, proposes a load–capacity cascading failure model based on improved percolation theory, integrating information flow overload and power system operational characteristics. This model analyzes the impact of overloaded links on the system's robustness.

1.4. Contributions

The contributions of this paper are summarized as follows:

- Considering the congestion state of communication network links, a dynamic transmission allocation model for information flow under three link states—normal, overload, and failure—is established. Metrics for communication system topology integrity and operational characteristics are proposed to assess system vulnerability in the event of faults.
- Considering the topology generation characteristics of actual power communication networks, an improved percolation theory is proposed. Communication nodes that are initially outside the largest connected component but have communication links to the control center are considered effective nodes. Power nodes that lose coupling with the communication network but remain self-consistent are also regarded as effective nodes.
- Considering the physical characteristics of coupled networks, the line flow betweenness indicator is utilized to measure the electrical characteristics of line power flows. A load–capacity distribution model based on flow betweenness is proposed for transferring the load of failed lines.

The rest of the paper is organized as follows. In Section 2, we construct the unidirectional dependency model of CPPS and provide a detailed description of the fault propagation process across the space considering communication link overload conditions within the coupled model. In Section 3, we propose two metrics of system robustness assessment to measure the cascading failure results. In Section 4, the numerical simulation is conducted to analyze the cascading failure and presents related discussions. Section 5 concludes the paper.

2. Methods

2.1. Modeling of Power–Communication Coupled Networks Based on Unidirectional Dependency

CPPS includes the modeling of the information network, power network, and coupling layer. The information network comprises the access layer, backbone layer, and core layer. The interdependent edges between power nodes and information nodes form the coupling network [27]. In the coupling network, the power network provides power support for the communication network, while the communication network offers 3C support for the power network. However, since communication nodes are widely equipped with backup power sources, the failure of coupled power nodes does not cause the failure of communication nodes due to power outage. The normal operation of power nodes relies on the monitoring and control of communication nodes. The failure of communication links will result in the control center being unable to receive the fault information of the power system in a timely manner, leading to the failure to promptly deal with power failures and expanding the scope of fault impact. Therefore, this paper primarily studies the model of the power network's dependence on the information network. Referring to the flowchart in Figure 2, the modeling process is detailed as follows:

(1) The physical power grid is abstracted as a graph $G_p(V_p, E_p)$ composed of power nodes V_p and transmission lines E_p. The set V_p includes physical equipment in the power grid such as power plants, substations, and loads.
(2) Divide the power grid into regions [28].
    a. The partitioning of communities is applied to the division of power network regions. In the power grid, the closer the distance between two buses, the smaller the line reactance and the larger its reciprocal (i.e., higher weight), indicating higher intimacy between the node pair. Thus, such nodes are more likely to be partitioned into the same region. The reciprocal of the line reactance is used as the weight for the non-zero elements of the power network adjacency matrix E.
    b. Using the Fast Newman method, the modified matrix E from the above steps is substituted for the original matrix E to calculate the modularity Q for the initial partitioning of the power network. The partitioning process must satisfy the following conditions: each region must contain at least one generator and one load; the number of regions must be less than the minimum of the number of generators and loads; and the subregions must achieve power balance.
    c. To achieve power balance within regions, the system's power flows on the lines are used as weight values based on the Prim algorithm. These weight values are used to determine the flow paths from generators to loads. Regions are then merged according to these paths and combined with the initial partitioning results to form the final zones.
(3) The number of nodes for each layer is determined: the access layer, backbone layer, and core layer. The access layer nodes are information collection nodes, with the number of communication nodes equal to the number of power nodes; the backbone layer nodes are routing equipment nodes, also considered as control nodes, with the number of nodes equal to the number of power grid partitions; and the core layer nodes are two control nodes representing the main and backup dispatch.
(4) The modeling of the information network:
    a. The connection between the power grid and the access layer: according to the abstract diagram of the power grid G_p, access layer nodes are connected in a one-to-one corresponding manner to make the topology diagram of the access layer consistent with the topology diagram of the power grid.
    b. The connection between the access layer and the backbone layer: the highest-degree nodes in each region of the access layer and the generator nodes are connected to the corresponding nodes in the backbone layer.
    c. The backbone layer nodes are connected internally according to the connection relationship of the power grid partitions.
    d. The connection between the access layer and the core layer: calculate the proportion q_i of the generator G_i output to the total system output, and connect it to the main and backup calls with a probability of q_i.
    e. All routing nodes are connected to the main and backup scheduling center nodes.
    f. The main and backup scheduling centers are connected.
(5) Abstract the information network to form G_c.
(6) Connect the access layer nodes with the power nodes to form unidirectional dependent edges.

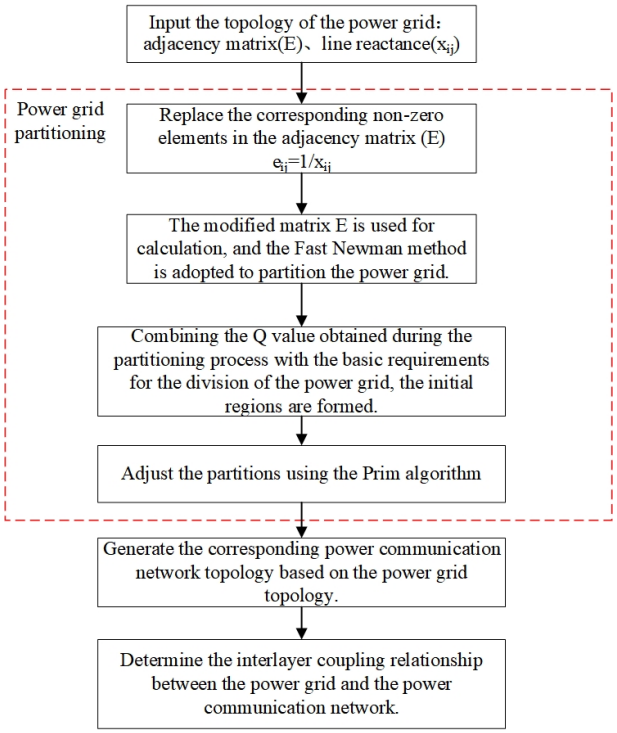


Figure 2. Flowchart of cyber–physical power system modeling.

Thus, the coupled networks are shown in Figure 3.

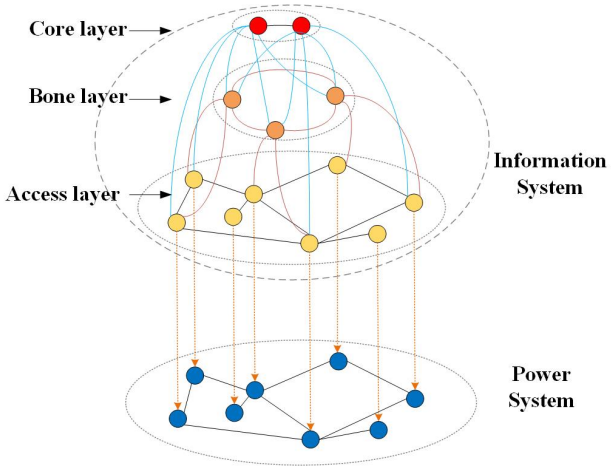


Figure 3. Topology diagram of the power grid dependent on the information network.

2.2. Cascade Failure Model Based on Percolation Theory Considering Communication Link Overloads

2.2.1. Nonlinear Flow Model for Links

The communication network's link information flow model employs a nonlinear capacity–load model that accounts for the overload state [29]. In complex networks, a higher node degree indicates more connections with other nodes. Therefore, the degree values of the endpoints are utilized to calculate the information flow of each access layer link. The higher the degree value of a link, the greater the information flow through it. The information flow ratio of a link is used as the link weight, and the overload coefficient δ describes the edge's capacity to handle additional information flow, as follows:

$$C_{c,ij} = L_{c,ij} + \beta L_{c,ij}^{\alpha} \quad (1)$$

$$C_{c,ij}^{\max} = \delta C_{c,ij} \tag{2}$$

$$W_{c,ij} = \begin{cases} \frac{L_{c,ij}}{C_{c,ij}} & e_{ij} \in E_C \\ 0 & e_{ij} \notin E_C \end{cases} \tag{3}$$

where

$$L_{c,ij} = w_{ij} = \left(k_i k_j\right)^{\theta} \tag{4}$$

where $L_{c,ij}$ represents the amount of information transmitted by the link and k_i and k_j denote the degrees of nodes i and j, respectively. θ is a parameter that adjusts the information flow. $C_{c,ij}$ is the capacity of the link, $C_{c,ij}^{\max}$ represents the maximum flow that the line can bear, and $W_{c,ij}$ is the weight of the edge e_{ij}. α and β are capacity coefficients. When $\alpha = 1$, the model is linear.

2.2.2. Improved Percolation Theory

Traditional percolation theory [30] is widely used to describe the structure, function, and resilience of network systems. Percolation models simulate link failure scenarios by gradually removing links from the network. As links are progressively removed, the reduction in the size of the largest connected subgraph can be used to measure the consequences of link failures. Thus, percolation theory is applicable to modeling cascading failures in cyber–physical power systems. However, traditional percolation theory only considers the largest connected subset in a single-layer network when determining the working node subset, without considering other scenarios. Directly applying it to model cascading failures in cyber–physical power systems makes it difficult to accurately simulate the failure process. Considering the information flow transmission characteristics of the information network, where the core layer and backbone layer do not directly correspond to the access layer, and the link status considers the overload situation, it is necessary to improve the classical percolation theory model. The failure model of the cyber–physical power system established in this paper is as follows:

- Communication system. For the links in the access layer, edges with weights greater than the overload coefficient are considered failed. For the information nodes in the access layer, nodes that cannot establish a path to control nodes are considered failed.
- Power system. A power load node must be connected to at least one generator node; otherwise, it is considered failed. Similarly, a generator node must be connected to at least one power load node; otherwise, it is considered failed. Nodes functioning as both generators and loads are considered self-consistent nodes.
- Interaction. Power nodes coupled with failed communication nodes will also fail. Additionally, power nodes coupled with communication nodes exiting transmission delays will fail with a certain probability P_{di}.

2.2.3. Load–Capacity Model of Power Network Based on Flow Betweenness

In previous studies, the shortest path propagation principle has commonly been used to investigate power transmission between buses in a power network. However, in reality, power flow in a power network does not follow only the path with the lowest impedance but rather propagates along all possible paths, adhering to Kirchhoff's Law.

To accurately reflect the role of each transmission line in power propagation and the varying impact of different generator–load pairs on each line, consider that in a given power network, each transmission line carries a varying proportion of transmission power $P(m, n)$ from generator m to load n. Consequently, each line plays a distinct role and has varying levels of significance in transmitting power $P(m, n)$. Since the power transmission paths for generator–load pairs that traverse each line differ, the line's significance within the entire

network is quantified using the flow betweenness index [26]. This index is computed by considering all generator–load pairs utilizing the line. The calculation formula is as follows:

$$F_{Bij} = \sum_{m \in G} \sum_{n \in L} \min(S_m, S_n) \frac{P_{ij,m} P_{ij,n} P_n}{P_{ij} P_{Ln} A^{-1}_{unm} P_{Gm}} \tag{5}$$

where G is the assembly of generation nodes and L is the assembly of consumer nodes. $min(S_m, S_n)$ is the weight of a single line's flow betweenness, which depends on the minimum value between the actual output of the generator m and the actual load n, reflecting the maximum available transmission power between m and n. $P_{ij,m}$ is the portion of the power flow on line e_{ij} originating from generator m. $P_{ij,n}$ is the portion of the power flow on line e_{ij} directed towards load n. P_n is the node power flow of n. P_{ij} is the active power through line e_{ij}. P_{Ln} is the active load at load node n. A^{-1}_{unm} contains the inverse-order distribution matrix elements. P_{Gm} is the active output of the generator node m.

Therefore, the power load $L_p(ij)$ on edge e_{ij} can be calculated as

$$L_p(ij) = F_{Bij} \tag{6}$$

Considering the capacity of links to handle power loads, we employ the load–capacity model proposed by Motter and Lai [31]. According to this model, the link capacity is directly proportional to its initial power load, as follows:

$$C_p(ij) = (1 + \gamma) L_p(ij) \tag{7}$$

where γ is the tolerance parameter.

2.2.4. Process of Cascading Failure

The information network has the capability to control power nodes. Consequently, faults occurring within the information network can propagate to the interconnected power network. This section details the dynamic process of cascading failures triggered by certain faulty communication links.

Based on the magnitude of information flow transmission in the communication network, the operational states of links are categorized into three types: normal, overloaded, and failed. To analyze cascading failures between coupled networks, some links in the communication network are randomly removed as initial faults. Referring to the flowchart in Figure 4, the detailed dynamic process of cascading failures triggered by specific damaged links is described as follows:

- Step 1: The set of failed links in the access layer communication network due to accidental failures or attacks is denoted as e_{ij}. The information flow on these failed links will be borne by the edges connected to the nodes of the failed links.
- Step 2: The process of distributing information flow on failed branches. This paper utilizes the principle of local redistribution of information flow, with the calculation formula provided as follows.

$$\Delta L_{c,ia} = L_{c,ij} \frac{L_{c,ia}}{\sum_{m \in \Omega_1} L_{c,im} + \sum_{n \in \Omega_2} L_{c,jn}} \tag{8}$$

where $L_{c,ij}$ is the initial information flow on a branch e_{ij}, $L_{c,ia}$ is the initial information flow on a branch e_{ia}, Ω_1 is the set of neighboring nodes of node i, and Ω_2 is the set of neighboring nodes of node j.

- Step 3: The process for determining the overloaded and failed states is as follows.

$$\begin{cases} W_{c,im} > \delta & \text{fail} \\ 1 < W_{c,im} < \delta \text{ and rand} > p_{im} & \text{overload} \\ 1 < W_{c,im} < \delta \text{ and rand} \leq p_{im} & \text{fail} \\ W_{c,im} \leq 1 & \text{normal} \end{cases} \tag{9}$$

where $rand \in (0,1)$.
Since each branch has different capacities to handle additional information flow, a distribution coefficient ω is introduced to characterize this property.

$$p_{ia} = \left(\frac{W_{c,ia} - 1}{\delta - 1} \right)^{\omega} \tag{10}$$

- Step 4: Distribution process for information flow on overloaded branches:

$$\Delta_{ak} = (L_{c,ia} - C_{c,ia}) T_{ak} \tag{11}$$

$$T_{ak} = \frac{C_{c,ak} - L_{c,ak}}{\sum_{e \in \Omega_i} (C_{c,ie} - L_{c,ie}) + \sum_{f \in \Omega_a} (C_{c,af} - L_{c,af})} \tag{12}$$

where Δ_{ak} is the distribution strategy for the overloaded branch, Ω_i is the set of neighboring nodes of node i with branches in a normal state, and Ω_a is the set of neighboring nodes of the node a with branches in a normal state.
- Step 5: Determine whether there are new failed branches. If new failed branches are detected, proceed to Step 2; otherwise, proceed to Step 6.
- Step 6: Count the sets of failed and overloaded links within the communication network. Update the effective set of communication links, and calculate the transmission delay increments for each communication node. Assess the failure conditions of communication nodes, remove nodes that cannot form a path to the control center, and update the set of effective communication nodes.
- Step 7: Control dependency analysis: Due to the one-to-one coupling between access layer communication nodes and power nodes, the failed communication nodes identified in Step 6 will cause corresponding power nodes to fail through dependent edges. Nodes are selected based on the delay probability of each communication node. If information from these nodes is not transmitted to the control center in a timely manner, their corresponding power nodes are considered failed. Mark the power nodes that have failed in this step.
- Step 8: Physical failure: First, remove the failed nodes in the power network. Next, based on the failure conditions for power system nodes, check if there are additional failed nodes among the remaining nodes and remove them if necessary. Recalculate the power load on the lines; if the redistributed load exceeds the capacity of any line, mark that line as faulty. Repeat this process until no more faulty nodes remain in the power network.
- Step 9: Output: Single cascading failure simulation ends. Output the data for both the power network and the communication network.

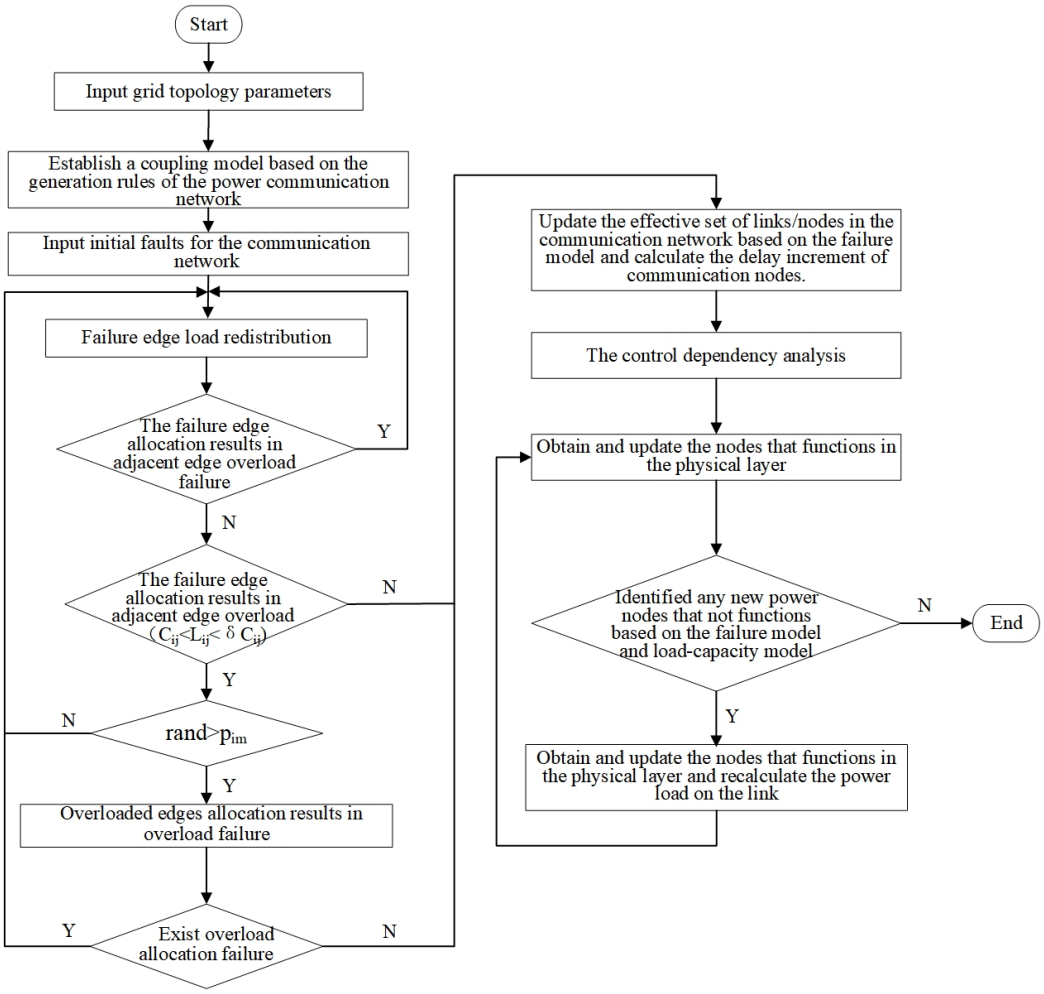


Figure 4. Flowchart of cascading failure process in cyber–physical power systems.

3. Evaluation Metrics of System Robustness to Cascading Failures

In CPPS, any faulty link has the potential to propagate through coupling relationships and evolve into cascading failures. To analyze the impact of initial failures on the system, we define evaluation metrics for both coupled networks based on network topology integrity and operational characteristics.

3.1. Metrics of Communication Network

In the communication network, we employ adjusted node/link survival rates to evaluate the communication network's topology integrity. Considering that communication links in an overloaded state have information flow exceeding their capacity—contrary to normal conditions—these links operate inefficiently and have a certain probability of transitioning to a failed state. Therefore, to distinguish between normal and overloaded links and more accurately represent the impact of the overloaded state on the system, the relative size of the largest connected component G_c is used to evaluate the communication links after adjustment [29]. The survival rate of nodes/links following failure is used to assess the topological integrity, as detailed below:

$$G_c = \frac{\sum_{h \in \Psi} s_h}{N_c} \tag{13}$$

where N_c is the number of nodes in the communication network's access layer. Ψ denotes the set of non-failed nodes. When the information links are in a normal state, $s_h = 1$.

However, when the information links are under overload conditions, s_h is calculated as follows:

$$s_h = \frac{\delta C_h - L_h}{\delta C_h - C_h} \tag{14}$$

Then, the adjusted node/link survival rate is

$$F_c = \frac{1}{2} * (\frac{V_c'}{V_c} + G_c) \tag{15}$$

Due to the disconnection of information links, the path from the access layer nodes of the communication network to the control center may change, resulting in communication delays. In this paper, the data communication delay is simplified and calculated as follows [32]: Data transmission in the communication network follows the shortest path principle, and each time it passes through a data node, the delay increases by one time unit τ. The delay time unit reflects the delay caused by data transmission and processing from the source node to the destination node in the communication network, including passing through each information node and the communication path to the next information node. Therefore, the transmission delay increment T caused by the communication network is calculated as follows:

$$T = \sum_{k \in L_c} T'_{\pi_c} + \sum_{k \in L_c} T_{\pi_c} \tag{16}$$

where T'_{π_c} and T_{π_c} are the delay of the transmission path of the same source–destination pairs after and before the cascading failure. L_c denotes the set of the shortest transmission paths for all source–destination pairs in the communication network.

3.2. Metrics of Power Network

In the physical power network, we use failure impact F_p to evaluate the influence of cascading failure [12], expressed as

$$F_p = \frac{1}{2} * (\frac{V_p'}{V_p} + \frac{E_p'}{E_p}) \tag{17}$$

where V_p' and E_p', respectively, are the number of failed nodes and links in the power network. V_p and E_p, respectively, are the total number of nodes and links in the physical power network.

4. Case Study and Discussion

In this section, taking the IEEE 39-bus system as an example, the power network topology is shown in Figure 5 and the original topology is divided into four regions based on zoning principles. Correspondingly, the power communication network, generated according to the aforementioned rules, is depicted in Figure 6. In this network, the access layer consists of 39 communication nodes, each corresponding to one of the 39 power nodes, which are used for uploading fault information and issuing dispatch instructions. The backbone layer includes four control nodes, each corresponding to its respective access layer region. The dispatch center is equipped with two control nodes.

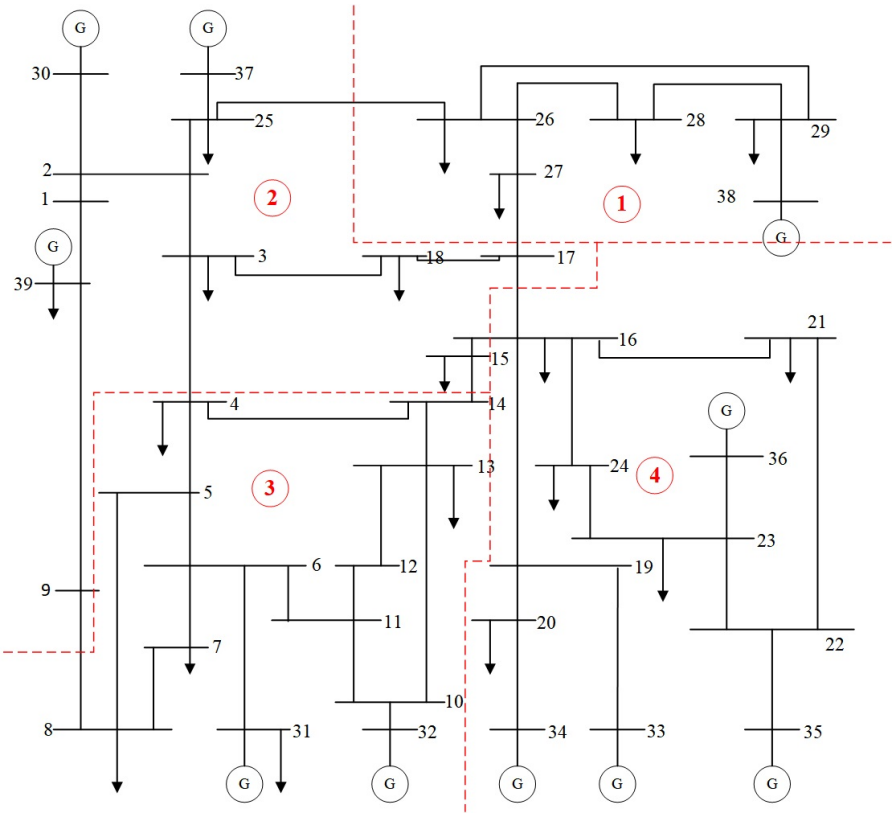


Figure 5. Partition diagram of IEEE 39-bus system.

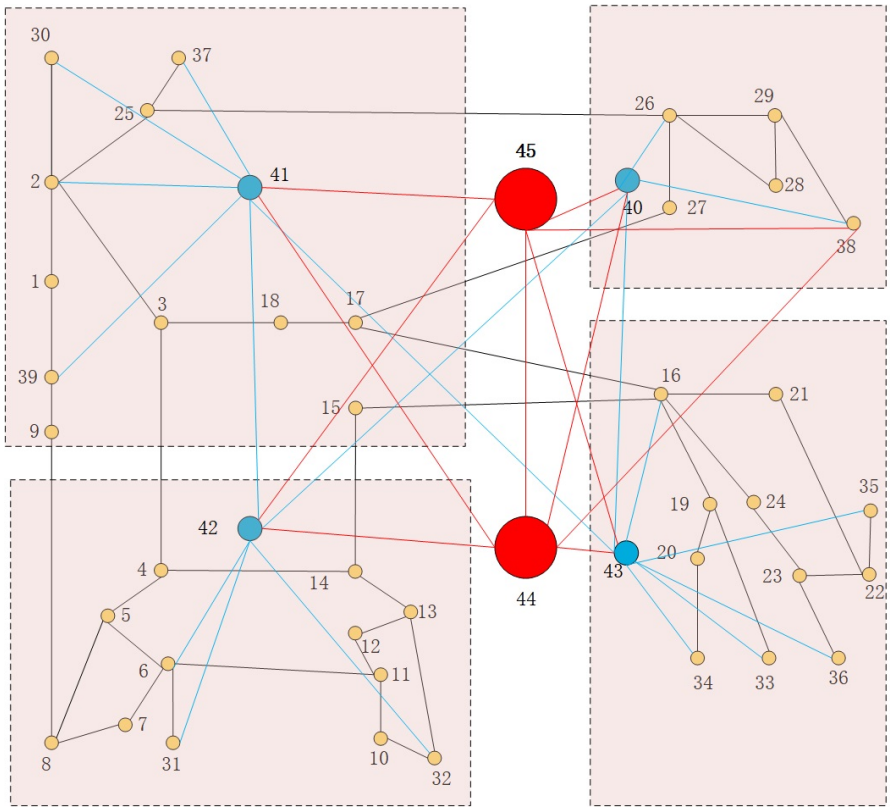


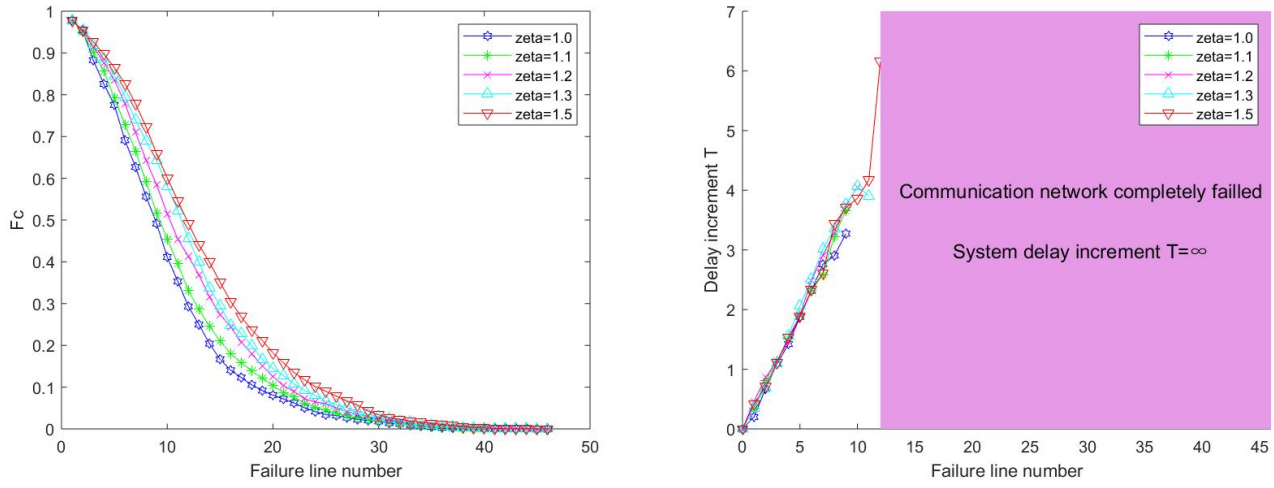
Figure 6. Topology diagram of information network generated by IEEE 39-bus system.

4.1. Impact of Overload Coefficient

To explore the impact of the overload coefficient on the robustness of the network, the number of initial link faults was varied, with each set of faulty links generated randomly. The simulations were independently run 50 times, and the average value was taken as the communication network metric. For clarity of presentation in charts, textual descriptions were used when the communication system completely failed, and the data set with the highest number of failed links that caused the complete failure of the communication system is selected for textual explanation. According to Figure 7a, as the number of initial faulty links increases, the survival rate of nodes/links in the communication network gradually decreases. When the overload capacity of links is not considered, the survival rate of nodes/links in the network is the lowest. When the overload coefficient is 1.2, the survival rate of nodes/links in the communication network significantly improves. However, it can be observed that when the overload coefficients are 1.3 and 1.5, the survival rate of nodes/links does not improve significantly.

Figure 7b shows that the CPPS under different overload coefficients exhibits a first-order percolation transition, with a similar overall trend. As the number of failed lines increases, the topology of the information layer is disrupted, and some nodes lose their data transmission paths. Changing the transmission path increases the system delay. When the overload coefficient is 1.5, the number of failed links that the communication network can withstand before complete collapse is the highest. As seen in Figure 7b and Table 1, there is a threshold for the initial number of failed links, near which the cascading failure effect expands to all communication links, leaving no transmission path between communication nodes and the control center. At this point, the communication system completely fails, making it impossible to control the power grid. Table 1 shows the specific thresholds for different overload coefficients.

Therefore, considering network construction costs, the overload coefficient is set to 1.3 according to Figure 7 and Table 1.



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

Figure 7. The robustness of the communication network and the increment in communication delay under different overload coefficients. (**a**,**b**) respectively show the relationship between the robustness of the communication network and the system delay with the increase in the number of communication link failures under different overload coefficients, while other parameters remain constant.

Table 1. The thresholds of the system under different overload coefficients.

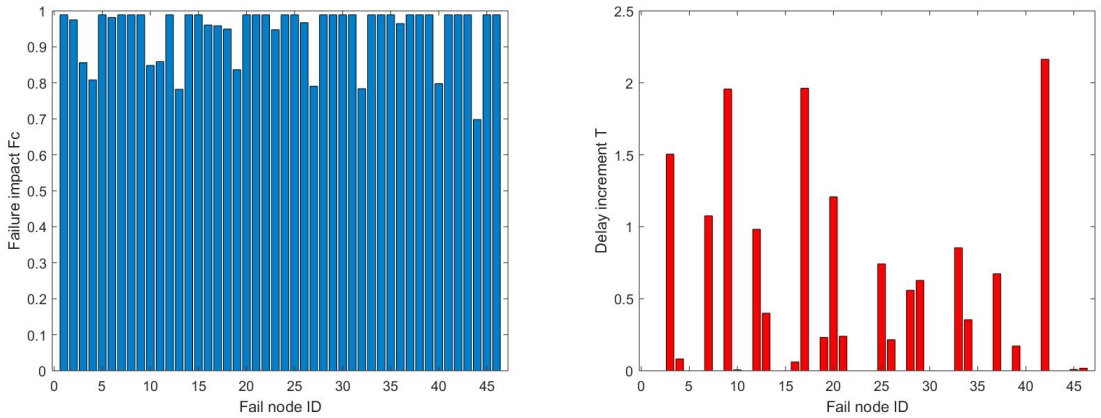
Overload Coefficient	The Threshold of Initial Failed Lines	Threshold Percentage %
1.0	9	19.56
1.1	9	19.56
1.2	10	21.74
1.3	11	23.91
1.5	12	26.09

4.2. Impact of Failed Links

In this section, we will analyze the impact on data transmission within the power and communication networks by sequentially removing each communication network link.

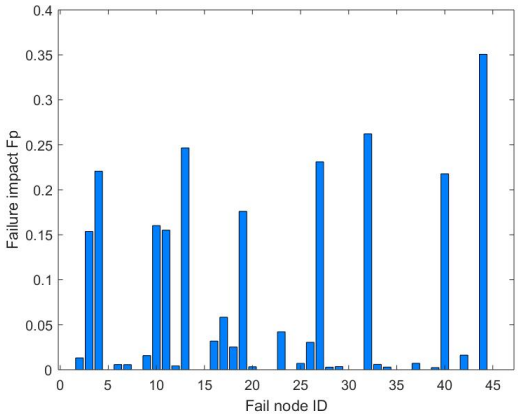
As shown in Figure 8, most of the faulty communication links have varying impacts on the topological integrity and operational characteristics of the CPPS. Links that cause significant system disruption are identified as critical links. Under N-1 fault conditions, all possible scenarios of system attacks are enumerated.

As shown in Figure 8, it can be observed that the failure of a single communication link can reduce the node and link survival rate in the communication network to as low as 69.69%, induce a system delay increment of 2.1635, and cause a 35.07% paralysis in the nodes and links of the power network. It is also noted that some link failures do not affect the rest of the communication and power networks beyond the faulty link. This is because, in cascading failures, the communication network considers the congested state of the links, redistributing the traffic on failed and overloaded links. Moreover, overloaded links can sustain operation for a short period before failing, thereby enhancing the system's robustness to some extent. Furthermore, the communication link that causes the maximum delay increment may not necessarily result in the lowest node/link survival rate in the communication network or the highest failure rate in the power network nodes/links. However, the failure rates of nodes/links in the communication and power networks are notably similar. For example, when link 42 fails, the highest delay increment of 2.1635 occurs. At this point, the survival rate of communication nodes/links is 98.91%, and the failure rate of power nodes/links is 3.46%. This is because the failure of link 42 changes the path from communication node 27 to the control center, from the original path 27→26→40→44 to 27→17→16→43→44, resulting in communication delay. The failure rate of communication node 27 is 36.06%. However, removing link 42 does not cause failure or overload in other links, so there are 45 operational links, and the number of effective communication nodes remains unchanged compared to before the fault. Due to the coupling relationship, the failure rate of power node 27 is also 36.06%, ultimately leading to the failure of power node 27.



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

Figure 8. *Cont.*



(**c**)

Figure 8. The impact of initial faults on the cyber–physical system. (**a**,**b**) show the impact of each communication link failure on the robustness and delay increment of the communication system, while (**c**) shows the impact of each communication link failure on the physical power grid.

4.3. Impact of Attack Strategies

The previous section analyzed the N-1 impact caused by the failure of a single communication link. As the number of faulty lines increases, different initial sets of faulty lines will have varying effects on the system. Therefore, based on structural and electrical characteristics, we employ both random line attack and critical line attack to remove communication network links one by one, analyzing the impact of different attack strategies on power and communication network data transmission.

The critical line attack strategy is derived from the initial power betweenness centrality importance ranking of the power network. Due to the uncertainty of the random line attack strategy, each faulty line in the random attack strategy is randomly generated, and the results are averaged after 50 independent runs.

As shown in Figure 9, compared to random attacks, the CPPS exhibits high vulnerability under attacks on critical lines. Under critical line attacks, even with the prior consideration of information flow redistribution due to link overload, the network still suffers significant damage. This is because the failure of critical lines disrupts the paths that transmit information to the control center, preventing the upload of power fault information. Consequently, the control center cannot respond, and the source nodes transmitting information in the communication network will cause the failure of coupled power nodes, thus expanding the fault scale and accelerating the propagation of cascading failures and system collapse.

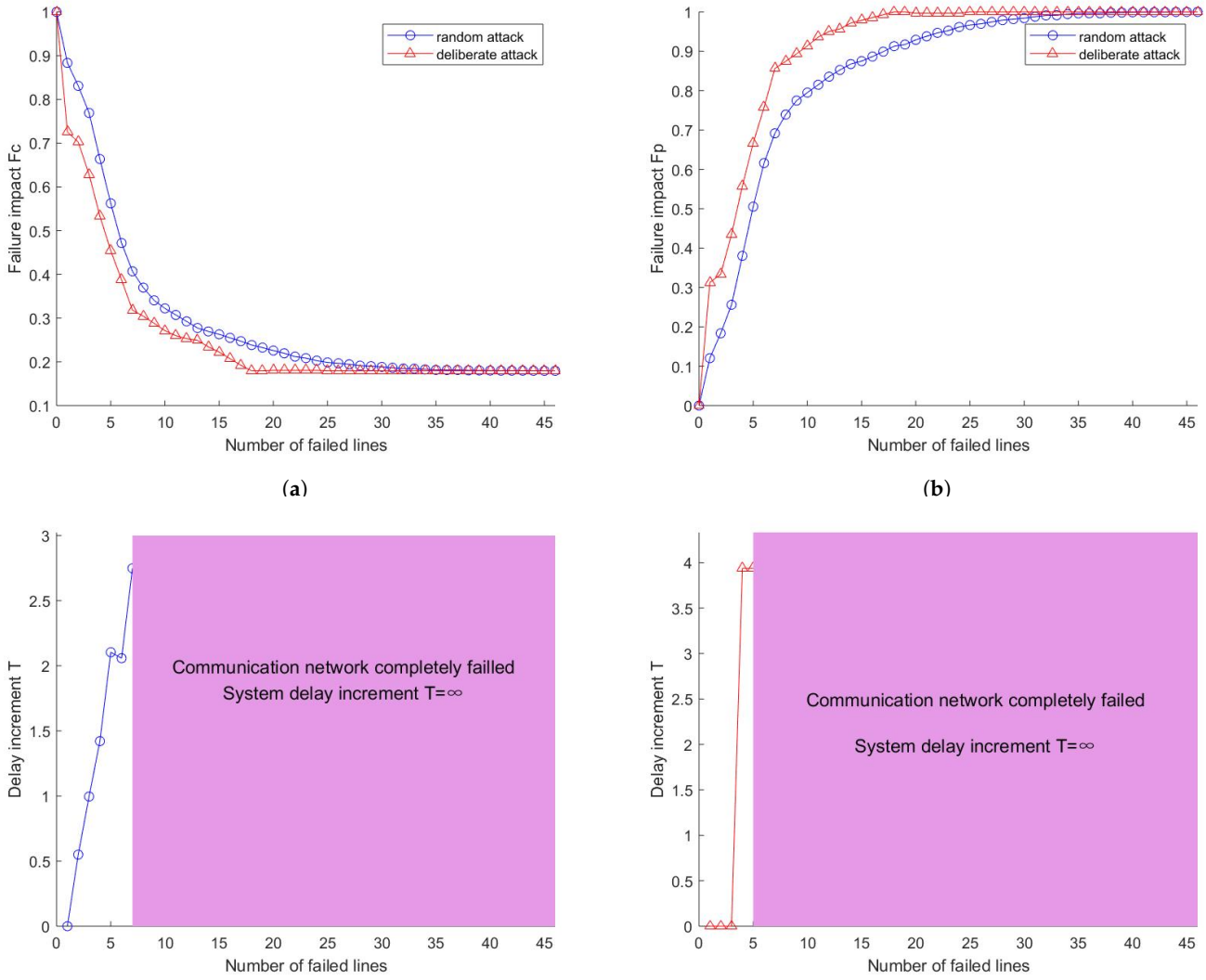


Figure 9. Impact of attack strategies on cyber–physical system. (**a**–**d**) illustrate the relationship between the evaluation metrics of the coupled network and the number of failed communication links under two different attack strategies (random attack and deliberate attack). Specifically, (**a**) shows the relationship between the survival rate of nodes/links in the power communication network and the number of failed communication links, (**b**) shows the relationship between the failure rate of nodes/links in the power communication network and the number of failed communication links. (**c**,**d**) respectively depict the trend of delay increment in the system under random attack and deliberate attack.

5. Conclusions

This paper investigates the dynamic propagation of cascading failures in CPPS. First, the coupling relationship between the communication network and the physical power network is analyzed, and the corresponding communication system's topology structure is generated based on the power network. Next, an overload distribution model for information flow is established within the communication network to address overloads resulting from failures. In the power network, a load–capacity distribution model based on power flow betweenness is established to handle the complexity of power flow calculations due to failures. To more accurately describe the node/line failures caused by power flow transfers due to faults, traditional percolation theory is improved to establish a failure propagation model for coupled networks.

Overall, this paper comprehensively considers both the electrical characteristics of the physical power grid and the information transmission characteristics of the power communication network when establishing the coupling model. It improves the coupled network model from both the topological structure and functional characteristics and constructs a cascading failure model that more accurately reflects real-world scenarios. Most existing studies are based on percolation theory and only analyze the cascading failure process of CPPS from the perspective of network topological integrity. This approach fails to realistically capture islanding phenomena in power system operations, thus necessitating an enhancement of percolation theory.

The simulations analyze the impact of faulty lines on the overload capacity of communication network links, data transmission, network survivability, and physical power network fault rates. The results indicate that even a small increase in link capacity can significantly enhance network robustness. Additionally, the system performance degrades when links at critical topological positions in the communication network are attacked, leading to increased system delay increments compared to random attacks and causing the information layer network to collapse earlier.

In summary, these findings offer valuable insights for future power network planning. However, these studies are based on post-fault conditions, considering only the interactions between nodes as either normal or complete failure to achieve power flow dispatch by the control center. In reality, the interaction process between power nodes and information nodes is extremely complex and often only qualitatively analyzed. Moreover, compared to power flow dispatch, scheduling information flow is relatively simple. Reducing the occurrence of faults from the source of the cyber side is a more efficient and reliable solution instead of establishing a cascading failure resistance mechanism on the power grid side. Therefore, our future research will focus on allocating critical information flows to reliable paths under conditions of communication link overload to ensure accessibility and reduce the probability of cascading failures caused by communication interruptions.

Author Contributions: Conceptualization, X.L. and Y.L.; Methodology, X.L. and Y.L.; Writing—original draft, X.L.; Writing—review and editing, Y.L. and T.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant 62062068 and Young Academic and Technical Leaders Program of Yunnan Province under Grant 202305AC160077.

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: Thanks for the help of Kunming AI Computing Center.

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

Abbreviations and Nomenclature

CPPS	Cyber–physical power system;
F_{ij}	Information flow demand on link e_{ij};
q_{ij}	The ratio of link capacity to link load;
ρ	The failure threshold of the link;
V_p	The sets of power nodes;
E_p	The sets of power transmission lines;
Q	The modularity value of the combined union of two communities;
q_i	The proportion of the generator i output to the total system output;
δ	The link's capacity to handle additional information flow;
k_i	The degrees of node i;
θ	The parameter that adjusts the information flow;
$L_{c,ij}$	The amount of information transmitted by the link e_{ij};
$C_{c,ij}$	The capacity of the link e_{ij};

$C_{c,ij}^{max}$	The maximum flow that the link e_{ij} can bear;
$W_{c,ij}$	The weight of the link e_{ij};
α	The capacity coefficient;
β	The capacity coefficient;
$min(S_m, S_n)$	The weight of a single line's flow betweenness;
$P_{ij,m}$	The portion of the power flow on line e_{ij} originating from generator m;
$P_{ij,n}$	The portion of the power flow on line e_{ij} directed towards load n;
P_{ij}	The active power through line e_{ij};
P_{Ln}	The active load at load node n;
A_{unm}^{-1}	The inverse-order distribution matrix elements;
P_{Gm}	The active output of the generator node m;
F_{Bij}	The flow betweenness of edge e_{ij} in the power network;
$L_p(ij)$	The power load on edge e_{ij} in the power network;
$C_p(ij)$	The capacity of edge e_{ij} in the power network;
γ	The tolerance parameter;
Δ_{ak}	The distribution strategy for the overloaded branch;
F_c	The adjusted node/link survival rate following failure in the communication network;
T	The transmission delay increment;
F_p	The failed node/link rate following failure in the power network.

References

1. Abdelmalak, M.; Venkataramanan, V.; Macwan, R. A survey of cyber-physical power system modeling methods for future energy systems. *IEEE Access* **2022**, *10*, 99875–99896. [CrossRef]
2. Li, Y.; Wang, B.; Wang, H.; Ma, F.; Ma, H.; Zhang, J.; Zhang, Y.; Mohamed, M.A. An Effective Node-to-Edge Interdependent Network and Vulnerability Analysis for Digital Coupled Power Grids. *Int. Trans. Electr. Energy Syst.* **2022**, *2022*, 5820126. [CrossRef]
3. Gao, X.; Peng, M.; Chi, K.T.; Zhang, H. A stochastic model of cascading failure dynamics in cyber-physical power systems. *IEEE Syst. J.* **2020**, *14*, 4626–4637. [CrossRef]
4. Liang, G.; Weller, S.R.; Zhao, J.; Luo, F.; Dong, Z.Y. The 2015 Ukraine blackout: Implications for false data injection attacks. *IEEE Trans. Power Syst.* **2016**, *32*, 3317–3318. [CrossRef]
5. Zhao, Q.; Qi, X.; Hua, M.; Liu, J.; Tian, H. Review of the recent blackouts and the enlightenment. In Proceedings of the CIRED 2020 Berlin Workshop (CIRED 2020), Online Conference, 22–23 September 2020; IET: Stevenage, UK, 2020; Volume 2020; pp. 312–314.
6. Wu, J.; Chen, Z.; Zhang, Y.; Xia, Y.; Chen, X. Sequential recovery of complex networks suffering from cascading failure blackouts. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 2997–3007. [CrossRef]
7. Peng, C.; Wu, J.; Tian, E. Stochastic event-triggered H∞ control for networked systems under denial of service attacks. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *52*, 4200–4210. [CrossRef]
8. Hu, S.; Yue, D.; Han, Q.L.; Xie, X.; Chen, X.; Dou, C. Observer-Based Event-Triggered Control for Networked Linear Systems Subject to Denial-of-Service Attacks. *IEEE Trans. Cybern.* **2020**, *50*, 1952–1964. [CrossRef]
9. Buldyrev, S.V.; Parshani, R.; Paul, G.; Stanley, H.E.; Havlin, S. Catastrophic cascade of failures in interdependent networks. *Nature* **2010**, *464*, 1025–1028. [CrossRef]
10. Huang, X.; Gao, J.; Buldyrev, S.V.; Havlin, S.; Stanley, H.E. Robustness of interdependent networks under targeted attack. *Phys. Rev. E—Stat. Nonlinear Soft Matter Phys.* **2011**, *83*, 065101. [CrossRef] [PubMed]
11. Cao, Y.Y.; Liu, R.R.; Jia, C.X.; Wang, B.H. Percolation in multilayer complex networks with connectivity and interdependency topological structures. *Commun. Nonlinear Sci. Numer. Simul.* **2021**, *92*, 105492. [CrossRef]
12. Chen, L.; Yue, D.; Dou, C.; Cheng, Z.; Chen, J. Robustness of cyber-physical power systems in cascading failure: Survival of interdependent clusters. *Int. J. Electr. Power Energy Syst.* **2020**, *114*, 105374. [CrossRef]
13. Pan, H.; Li, X.; Na, C.; Cao, R. Modeling and analysis of cascading failures in cyber-physical power systems under different coupling strategies. *IEEE Access* **2022**, *10*, 108684–108696. [CrossRef]
14. Jianfeng, D.; Jian, Q.; Jing, W.; Xuesong, W. A vulnerability assessment method of cyber physical power system considering power-grid infrastructures failure. In Proceedings of the 2019 IEEE Sustainable Power and Energy Conference (iSPEC), Beijing, China, 21–23 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1492–1496.
15. Tian, M.; Dong, Z.; Gong, L.; Wang, X. Line hardening strategies for resilient power systems considering cyber-topology interdependence. *Reliab. Eng. Syst. Saf.* **2024**, *241*, 109644. [CrossRef]
16. Chen, L.; Yue, D.; Dou, C.; Xie, X.; Li, S.; Zhao, N.; Zhang, T. Impact of cascading failure on power distribution and data transmission in cyber-physical power systems. *IEEE Trans. Netw. Sci. Eng.* **2023**, *11*, 1580–1590. [CrossRef]
17. Zhong, J.; Zhang, F.; Yang, S.; Li, D. Restoration of interdependent network against cascading overload failure. *Phys. A Stat. Mech. Its Appl.* **2019**, *514*, 884–891. [CrossRef]

18. Ding, D.; Wu, H.; Yu, X.; Wang, H.; Yang, L.; Wang, H.; Kong, X.; Liu, Q.; Lu, Z. Vulnerability Assessment of Cyber Physical Power System Based on Improved Cascading Failure Model. *J. Electr. Eng. Technol.* **2024**, 1–12. [CrossRef]
19. Wang, S.; Gu, X.; Chen, J.; Chen, C.; Huang, X. Robustness improvement strategy of cyber-physical systems with weak interdependency. *Reliab. Eng. Syst. Saf.* **2023**, *229*, 108837. [CrossRef]
20. Ghasemi, A.; de Meer, H. Robustness of interdependent power grid and communication networks to cascading failures. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 1919–1930. [CrossRef]
21. Cao, R.; Dong, X.; Wang, B.; Liu, K. Discussion of protection and cascading outages from the viewpoint of communication. In Proceedings of the 2011 International Conference on Advanced Power System Automation and Protection, Beijing, China, 16–20 October 2011; IEEE: Piscataway, NJ, USA, 2011; Volume 3, pp. 2430–2437.
22. Zhang, G.; Shi, J.; Huang, S.; Wang, J.; Jiang, H. A cascading failure model considering operation characteristics of the communication layer. *IEEE Access* **2021**, *9*, 9493–9504. [CrossRef]
23. Gao, X.; Peng, M.; Chi, K.T. Robustness analysis of cyber-coupled power systems with considerations of interdependence of structures, operations and dynamic behaviors. *Phys. A Stat. Mech. Its Appl.* **2022**, *596*, 127215. [CrossRef]
24. Wang, F.; Li, D.; Xu, X.; Wu, R.; Havlin, S. Percolation properties in a traffic model. *Europhys. Lett.* **2015**, *112*, 38001. [CrossRef]
25. Saberi, M.; Hamedmoghadam, H.; Ashfaq, M.; Hosseini, S.A.; Gu, Z.; Shafiei, S.; Nair, D.J.; Dixit, V.; Gardner, L.; Waller, S.T.; et al. A simple contagion process describes spreading of traffic jams in urban networks. *Nat. Commun.* **2020**, *11*, 1616. [CrossRef]
26. Liu, W.; Liang, C.; Xu, P.; Dan, Y.; Wang, J.; Wang, W. Identification of critical line in power systems based on flow betweenness. *Proc. CSEE* **2013**, *33*, 90–98.
27. Wang, T.; Sun, C.; Gu, X.; Qin, X. Modeling and vulnerability analysis of electric power communication coupled network. *Proc. CSEE* **2018**, *38*, 3556–3567.
28. Newman, M.E. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E—Stat. Nonlinear Soft Matter Phys.* **2006**, *74*, 036104. [CrossRef]
29. Chen, C.Y.; Zhao, Y.; Gao, J.; Stanley, H.E. Nonlinear model of cascade failure in weighted complex networks considering overloaded edges. *Sci. Rep.* **2020**, *10*, 13428. [CrossRef]
30. Stauffer, D.; Aharony, A. *Introduction to Percolation Theory*; Taylor & Francis: Abingdon, UK, 2018.
31. Motter, A.E.; Lai, Y.C. Cascade-based attacks on complex networks. *Phys. Rev. E* **2002**, *66*, 065102. [CrossRef]
32. Han, Y.; Guo, C.; Zhu, B.; Xu, L. Model cascading failures in cyber physical power system based on improved percolation theory. *Autom. Electr. Power Syst.* **2016**, *40*, 30–37.

electronics

MDPI

Article

A Practical Security Assessment Methodology for Power System Operations Considering Uncertainty

Nhi Thi Ai Nguyen [1], Dinh Duong Le [1,*], Van Duong Ngo [1], Van Kien Pham [1] and Van Ky Huynh [2]

[1] Faculty of Electrical Engineering, The University of Danang—University of Science and Technology, Danang 550000, Vietnam; ntanhi@dut.udn.vn (N.T.A.N.); nvduong@dut.udn.vn (V.D.N.); pvkien@dut.udn.vn (V.K.P.)
[2] The University of Danang, Danang 550000, Vietnam; hvky@ac.udn.vn
* Correspondence: ldduong@dut.udn.vn

Abstract: Today, renewable energy sources (RESs) are increasingly being integrated into power systems. This means adding more sources of uncertainty to the power system. To deal with the uncertainty of input random variables (RVs) in power system calculation and analysis problems, probabilistic power flow (PPF) techniques have been introduced and proven to be effective. Currently, although there are many techniques proposed for solving the PPF problem, the Monte Carlo simulation (MCS) method is still considered as the method with the highest accuracy and its results are used as a reference for the evaluation of other methods. However, MCS often requires very high computational intensity, and this makes practical application difficult, especially with large-scale power systems. In the current paper, an advanced data clustering technique is proposed to process input RV data in order to the decrease computational burden of solving the PPF problem while upholding an acceptable level of accuracy. The proposed method can be effectively applied to solve practical problems in the operating time horizon of power systems. The developed approach is tested on the modified IEEE-300 bus system, indicating good performance in reducing computation time.

Keywords: probabilistic power flow; power system; renewable energy

Citation: Nguyen, N.T.A.; Le, D.D.; Ngo, V.D.; Pham, V.K.; Huynh, V.K. A Practical Security Assessment Methodology for Power System Operations Considering Uncertainty. *Electronics* **2024**, *13*, 3068. https://doi.org/10.3390/electronics13153068

Academic Editor: Ahmed Abu-Siada

Received: 26 June 2024
Revised: 31 July 2024
Accepted: 31 July 2024
Published: 2 August 2024

1. Introduction

During the operation of a power system, operating mode parameters such as the voltage at the buses, power transmitted through the branches, etc., need to be regularly calculated to assess the security of the system by comparing the parameters with their allowable limits. In the event of a safety risk, reasonable solutions should be proposed to resolve it. Deterministic power flow (DPF) is one of the essential tools for power system operation and planning. Nevertheless, during the computation process, the traditional approach uses fixed values of nodal power injections (from power generation, load, etc.) and the known grid structure so that sources of uncertainty from these factors are not considered. This is the main limitation of the traditional power flow (PF) method [1].

To overcome the above-mentioned disadvantage, PPF was proposed and has become a very effective calculation tool. The load, the power generation from a plant, and the operation of an element such as a line, transformer, etc., can follow certain rules of probability. In particular, for today's power systems, when additional RESs such as solar and wind power, etc., are connected to the system, modeling renewables' intermittency is very challenging. The intermittency often changes very quickly and stochastically, increasing the uncertainty level in the system. Therefore, a calculation method is required to be able to integrate uncertainties into the calculation process. By using PPF methods, the outputs, i.e., the voltage at buses, PF transmitted on branches, etc., also change randomly according to a law of probability distribution [1]. The PPF analysis allows us to appraise the probability of line overloading, the probability of over-/under-voltage, etc. From there, depending on the

characteristics of the system and the severity of the violation, the operator could consider and suggest appropriate solutions to improve the system security.

The PPF approach was first introduced by Borkowska in 1974 [2] and, since then, several research works for PPF have been proposed around the world. Generally, methods for calculating the PF using the PPF technique can be classified into three main categories, i.e., numerical, analytical, and approximation approaches.

The analytical approach [3–6] makes use of algorithms and analytical techniques such as the convolution and cumulant techniques. Applying these analytical techniques combined with the relationship of the input and output of a PPF problem allows for determination of the distribution function of the output RVs such as the power transmission on the line, nodal voltage, phase angle, etc., according to the system parameters, e.g., the total line impedance, total transformer impedance, etc., and probability distributions of the input RVs of the load and power generation from traditional generators and RESs, as well as the operating status of the devices. The relationship between the input and output of the PF calculation problem is non-linear. Nevertheless, the analytical method works well with a linear relationship between the input and output of the problem. Therefore, the relationship, firstly, needs to be linearized using an expansion technique, e.g., Taylor expansion. One of the outstanding advantages of the analytical approach is that it can give very fast results. Among the cumulant and convolution approaches, the convolution approach is more computationally intensive than the cumulant one. Hence, currently, the cumulant approach is more popular than the convolution approach. To achieve the distribution functions for the output RVs, the cumulant approach is often used simultaneously with expansion techniques such as Gram–Charlier or Cornish–Fisher expansion [4]. Owing to the advantage of fast computation, the analytical approach could be applied for a large-scale power system in practice. Nevertheless, the analytical approach has some drawbacks. Firstly, the accuracy of the analytical approach is significantly affected by the use of techniques that linearize the input and output relationship, especially when the input RV changes over a wide range, for example, in the case of RESs. Secondly, the analytical approach uses expansion techniques that can perform well in the case of the distribution functions of the input RVs being either Gausian distribution or close to Gausian distribution. In fact, the distribution functions of the input RVs of the PPF problem for a power system, in practice, often follow non-Gausian distribution, so the achieved results will be limited. To be able to integrate discrete distribution functions of input RVs into the calculation process, the Von Mises method is proposed [1].

The typical approach for the group of approximation ones in calculating PPF is the point estimate approach [7,8]. In this approach, the input RV is decomposed into a sequence of value and weight pairs. Next, the moment of the output RV is computed as a function of the input RV and then the output RV distribution function is obtained. The point estimate approach can provide relatively fast results. Moreover, different from the analytical approach, this approach uses the non-linear relationship between the input and output of the PF computation problem. However, the main limitation of the point estimate approach is that its accuracy decreases as the order of the moment increases. Another drawback is that the computation time required increases significantly as the number of input RVs increases.

A typical numerical approach is MCS [9–14]. In MCS, the input RVs are sampled and then the DPF calculation is carried out for all samples. It repeats the simulation with a very large number of samples to obtain a highly accurate result. MCS uses the non-linear relationship between the input and output of the PF problem, like the traditional approach. The main advantage of the MCS approach is that it gives very accurate and reliable results. Moreover, the probability distributions of the input RVs in MCS are easy to represent. However, the biggest drawback is that the calculation volume is heavy and the calculation time is relatively long, thus making it difficult to apply for a practical large-scale power network. To reduce the computational burden of the MCS method, several clustering algorithms are proposed to reduce the number of samples, and then DPF is run for each cluster instead of running it for all samples. In [15], a PSO algorithm is proposed to use for

the clustering task, while K-means is used in [16,17]. Each clustering algorithm has its own weaknesses. For PSO, its iterative convergence rate is modest, and it often becomes stuck at local optimums in dealing with high-dimensional datasets. For the K-means algorithm, it is sensitive to the choice of k and it is difficult to find the optimal k for a given dataset. Moreover, K-means does not scale well to large problems, which is why, in the present study, this problem is focused on.

From the above analysis, it can be seen that each PPF approach has its own characteristics, advantages, and disadvantages.

In addition to the above overview of PPF, to solve various problems related to uncertainty, recent advancements have also been found. In [18], a two-stage robust coordinated dispatch method for multi-energy microgrids is developed to alleviate all of the negative effects of diverse uncertainties from wind power and loads. The interval method is employed in [18] to characterize the uncertainties. A committed carbon emission operation region (CCEOR) of integrated energy systems (IESs) is proposed in [19]. The developed method converts the proposed uncertain non-linear CCEOR model into a deterministic mixed-integer convex CCEOR model. In [20], to take into account different types of uncertainties from the outcomes of disasters, extreme events, loads, and renewable generation, both the pre-restoration and real-time stage measures are coordinated via a two-stage stochastic programming method.

The main contributions of this paper are summarized as follows: (1) The core objective of this study is to develop an approach to calculating PPF that ensures a certain level of accuracy compared to MCS but which must give very fast results close to "real time" operation of the power system. In order to exploit the advantages of the accuracy of the MCS method while reducing the time-consuming, and volume of, calculation, a real time clustering technique combined with MCS in PPF calculation is proposed. The clustering technique applied is simple but effective and suitable for practical application. The large number of samples of input RVs of the PPF calculation problem using MCS is significantly and effectively reduced, so the PPF calculation gives fast results. Thanks to this outstanding feature, the PPF approach can be applied to large power systems in practice and to the operational time frame. (2) In addition, the discussion on the application of PPF methods in power system calculation and analysis is also presented in detail in this article. This provides a clearer and more intuitive picture of the application of PPF analysis in both the planning and operation of power systems. The current limitations of PPF methods in general are also pointed out to provide topics for future research.

The rest of this paper is structured as follows. Section 2 presents the developed methodology, while the results obtained by the developed approach are discussed in Section 3. In Section 4, further discussion on the applicability of various PPF methods in power system analysis problems is given. Concluding remarks are provided in Section 5.

2. Methodology

2.1. Real Time Clustering Technique

Clustering is the division of data into a number of groups so that data points in the same group have similar characteristics to each other and are dissimilar to data points in other groups. It is basically the task of dividing data in a dataset on the basis of similarities and differences between them. Among clustering techniques, K-means is known as the most popular one and is applied in all fields due to its simplicity, efficiency, scalability, and ease of implementation. It can handle large datasets effectively, making it a practical choice for numerous applications. A comprehensive review of the application of K-means clustering in modern power systems is presented in [21]. The K-means clustering algorithm is a type unsupervised machine learning that divides the unlabeled dataset into k different clusters by an iterative algorithm.

The K-means algorithm is implemented as follows:

Step 1: Randomly choose k points or centroids from considered data to initialize the groups or clusters;

Step 2: For each point in the dataset, compute the distance between the point and each of the k centroids; assign each point to its closest centroid to form k clusters;

Step 3: Replace the centroid for each cluster with the mean of all data points assigned to the cluster;

Step 4: Repeat Steps 2 and 3 until the centroids no longer change significantly or after a pre-selected maximum number of iterations. The outputs obtained are the last cluster centroids and the data points assigned to clusters.

Although the K-means method has many advantages, it has some disadvantages that can affect its applicability and performance. The K-means algorithm is not considered to have good scalability for large problems. It is sensitive to the choice of k and it is difficult to find the optimal k for a given dataset. K-means converges to a local minimum, so different initializations will result in different results.

K-means can be very time-consuming with large datasets. For a dataset including n data points, it needs to be run O(nkT) times to calculate the distances between the n data points and each of the k centroids (T is the number of iterations) [22]. In the K-means algorithm, each iteration takes a time proportional to k and n. This explains why the K-means algorithm has poor scalability. The running time will increase with an increasing n or k, or both. Therefore, its efficiency can be significantly enhanced by decreasing the runtime related to n. To deal with the problem of poor scalability, in [22], the authors develop a K-means-lite approach that can obtain the aimed-for centroids in O(1) time with respect to n and exhibits and improved speed-up factor as k and n increase. The accuracy has also been shown to increase. The statistical inference technique is used, in which the k centroids are calculated using a few small samples, instead of repeated exhaustive comparison between centroids and data points. This idea comes from an intuitive extension of the classical central limit theorem. In particular, its use does not need special data structures, does not need to keep distances computed in memory, and does not require repeated exhaustive assignments. It is demonstrated that the use of K-means-lite obtains a drastic efficiency gain and can solve large datasets in real time; it is called advanced data clustering (ADC) in this paper [22].

2.2. Representation of Input Uncertainties

In this paper, to account for the uncertainties regarding the power outputs of generators, loads, etc., and the parameters of components, they are represented by probabilistic distributions. Based on their historical data, the distributions can be estimated. They can also be provided by a forecast technique, especially in solving operational problems.

- Wind generation

For wind speed modeling, the Weibull distribution [23] is commonly used. Its probability density function (PDF) is represented as:

$$f(v) = \frac{h}{c} \cdot \left(\frac{v}{c}\right)^{h-1} \cdot \exp\left[-\left(\frac{v}{c}\right)^{h}\right] \quad (1)$$

where h: the shape parameter; c: the scale parameter; v: wind speed.

The wind turbine characteristic curve can be estimated by wind power–wind speed pairs measurement data [24]. It can also be modeled by a piecewise function as follows:

$$P_{wo}(v) = \begin{cases} 0 & v \le v_{ci} \text{ or } v > v_{co} \\ P_w \dfrac{v - v_{ci}}{v_r - v_{ci}} & v_{ci} < v \le v_r \\ P_{wr} & v_r < v \le v_{co} \end{cases} \quad (2)$$

where, v_{ci}, v_{co}, and v_r are the cut-in, cut-out, and rated wind speed, respectively; P_{wr} and P_{wo} are the rated power and output of the wind generation, respectively.

- Solar generation

The solar radiation distribution can be estimated by its observed data. It is also usually represented by a Beta distribution [25], as:

$$f(r) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \cdot \left(\frac{r}{r_{\max}}\right)^{\alpha-1} \cdot \left(1-\frac{r}{r_{\max}}\right)^{\beta-1} \tag{3}$$

where r and r_{max} are the real and maximum solar radiations, respectively; α and β are two main parameters of the distribution; $\Gamma(\cdot)$ is the well-known Gamma function.

$$P_{vo}(r) = \begin{cases} P_{vr}\dfrac{r^2}{r_c r_{std}} & r < r_c \\ P_v\dfrac{r}{r_{std}} & r_c \le r \le r_{std} \\ P_{vr} & r > r_{std} \end{cases} \tag{4}$$

where r_c is the radiation at a certain point; r_{std} is the standard radiation (corresponding to the standard environment); P_{vr} and P_{vo} are the rated and output powers of the photovoltaic unit, respectively. Solar generation is commonly required to operate in the unity power factor mode, i.e., its reactive power is equal to zero.

- Load

The uncertainty of each load is usually represented by a Gaussian or normal distribution [1]. The normal distribution function is a continuous function and one of the most commonly used functions in most fields.

The PDF of a normal distribution is as follows:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{5}$$

in which μ and σ are the expectation (average value) and the standard deviation, respectively.

The cumulative distribution function (CDF) of the normal distribution function is calculated as follows:

$$F(x) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{x} e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt \tag{6}$$

For modeling a load, the expected value (mean) is its base power while the standard deviation is assumed to be equal to a certain percentage, e.g., 10%, of the mean.

In addition to the popular probability distributions, shown above, which are very suitable for representing uncertainties from RESs and loads in power systems, currently, in the fields of probability and statistics, there are several other probability distributions and density functions used to incorporate uncertainties into the PPF problem. In other words, the fact that we assume the above distribution functions for RESs and loads does not lose the generality of the use of the PPF method developed in this study.

2.3. Advanced Data Clustering-Based Probabilistic Power Flow

The flowchart of the proposed approach, i.e., the advanced data clustering-based probabilistic power flow (ADCPPF), used for probabilistic security assessment is shown in Figure 1. In the flowchart in Figure 1, the main part that helps significantly improve the calculation time is in the ADC block.

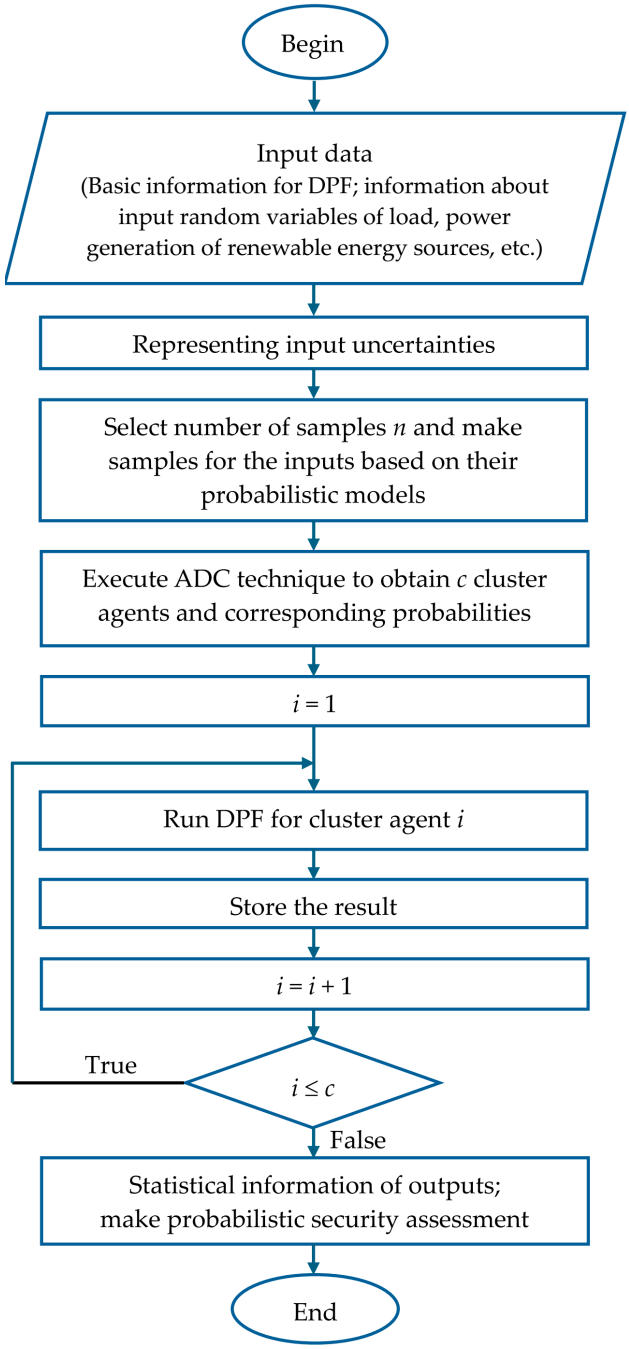


Figure 1. Flowchart of the proposed probabilistic security assessment.

3. Tests and Results

The application of the developed approach is illustrated on a modified IEEE 300-bus system. The information needed for DPF analysis of the system, i.e., the electrical network diagram, bus, branch, generator data, is given in [26]. The system includes 300 buses, 409 branches, 195 loads, and 69 generators. In this test, uncertainties from both loads and RESs are considered. The system is modified by adding 10 solar photovoltaic and 8 wind power plants to buses, as shown in Tables 1 and 2, respectively.

Table 1. Information on Beta distributions of solar photovoltaic power.

Bus	Rated Power (MW)	α Parameter	β Parameter
196	50	2.5	8
198	80	1.6	9
203	40	1.2	6
204	60	2.2	8
215	70	3.2	7
217	50	3.5	10
221	35	4.2	11
229	90	2.8	8
245	65	1.9	7
246	95	3.1	8

Table 2. Information on Weibull distributions of wind power.

Bus	Rated Power (MW)	Scale Parameter	Shape Parameter
118	90	10	2.4
121	80	15	1.6
126	100	11	2.4
142	50	14	1.5
154	40	20	2.2
156	60	28	1.8
159	70	16	1.7
161	95	12	2.3

For the sake of simplicity, but without the loss of generality, the uncertainties of loads and RESs are assumed to be known by forecast techniques. The uncertainty of each load is modeled by a normal distribution with an expected value equal to the base value and standard deviation equal to 10% of the expected value. The solar photovoltaic power uncertainty at each plant is assumed to follow Beta distribution, with its parameters given in Table 1. The distributions are also assumed to be correlated with a correlation coefficient of 0.7. In this study, it is assumed that the simulation scenario occurs during daylight hours, with potential conditions for some solar photovoltaic power generation. Additionally, we assume that the loads and solar and wind power plants are not under abnormal weather conditions (e.g., extreme heat wave, large-scale winter storm, and hurricanes), extremely rare phenomena (e.g., solar eclipse), and catastrophes (e.g., war, earthquakes, and tsunamis). Some of these abnormal events are extremely challenging to forecast with low standard deviation. These abnormal and rare events can also be described by appropriate distribution functions and included in the PPF problem. However, this issue is out of the scope of the current study and is intended to be considered in future studies. Similarly, the uncertainty of the power output at each wind farm is assumed to have Weibull distributions, with the parameters shown in Table 2. The distributions are correlated with a correlation coefficient of 0.8.

In the current test, the base power of 100 MVA is used. The MCS results are used as the reference to evaluate the results obtained by other methods. All tests are executed in Matlab (R2015b) on an Intel Core i5 CPU 2.53 GHz and 4.00 GB RAM PC.

PPF computation is performed to achieve all results of interest of the output RVs in terms of PDFs and/or CDFs. For the purpose of illustration, the distributions of a number of selected output RVs are shown. Figure 2 plots the CDFs of real PF through branch 2–8

(i.e., denoted as $P_{2–8}$), while that of the voltage at bus 89 (i.e., denoted as V_{89}) is given in Figure 3. It can be seen from the figures that the developed ADCPPF approach can match well with the curves from MCS, indicating the good performance of the ADCPPF approach.

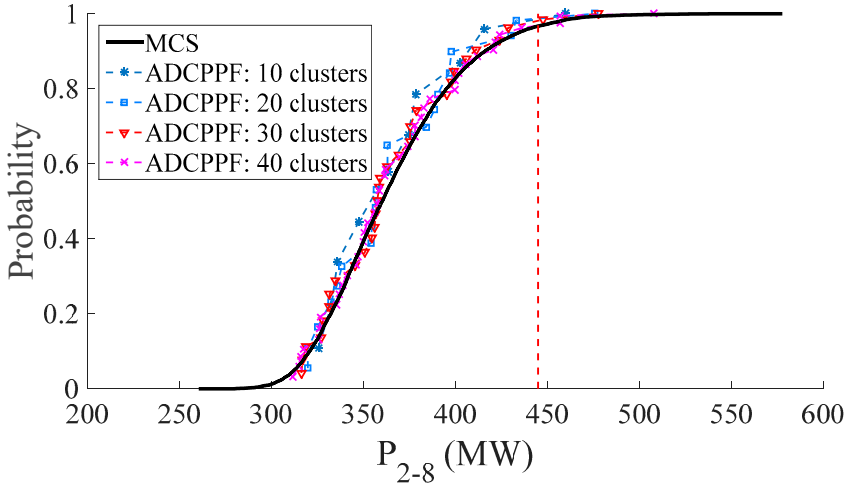


Figure 2. CDFs of active PF through branch 2–8.

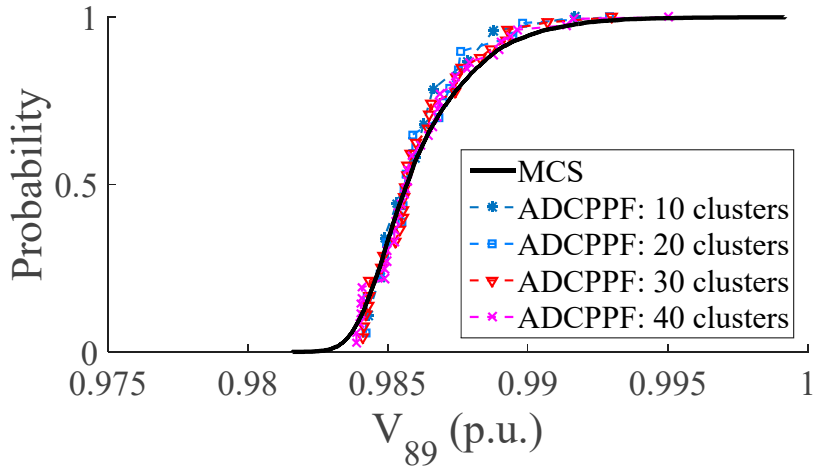


Figure 3. CDFs of voltage at bus 89.

Because it is difficult to observe clearly when plotting all the results on the same figure, only the results corresponding to the ADCPPF method are depicted. However, to demonstrate its effectiveness, we also compare the results of PPF using ADC with K-means clustering.

As discussed in Section 2.1, K-means clustering can be very time-consuming with large datasets due to the poor scalability problem. Different from K-means, the K-means-lite approach was developed in [16] and can give the results very quickly. Its accuracy has also been shown to increase in comparison with the K-means technique. Therefore, in this test, we do not focus on proving the accuracy of the ADC method compared to the K-means method. Instead, the processing time performance is focused on, thereby indicating that the proposed approach has good applicability for solving problems in the time frame of power system operation. It is clearly shown in Table 3 that the ADCPPF approach can give the result in a few seconds, in comparison to the hundreds of seconds needed by MCS. In particular, as previously mentioned, for a dataset including n data point, K-means clustering needs an amount of time equal to O(nkT) to run (in which T is the number of iterations). The running time of K-means will sharply increase with an increasing n and/or

k. The modified IEEE 300-bus system is a large-scale system, so K-means-based PPF runs very hard and takes a long time.

Table 3. Execution time comparison.

Method	Time (s)
MCS	725
ADCPPF with 5 clusters	2.64
ADCPPF with 10 clusters	2.79
ADCPPF with 20 clusters	2.98
ADCPPF with 30 clusters	3.26
ADCPPF with 40 clusters	3.47
ADCPPF with 50 clusters	3.72
ADCPPF with 70 clusters	4.21
K-means based PPF with 5 clusters	140
K-means based PPF with 10 clusters	420

From Figures 2 and 3 and Table 3, the accuracy of ADCPPF increases (i.e., intuitively, the corresponding curve in Figures 2 and 3 follows the MCS curve more closely) and the time required to execute the method also increases with an increasing number of clusters. Comparing MCS using K-means and ADCPPF, K-means is challenging in this case and causes the calculation time to increase by much more than when using ADCPPF. Through the above analysis, it is shown that the ADCPPF method has the advantage of both relatively high accuracy and a significantly reduced execution time.

PPF can provide distributions for output RVs that are good for power system security assessment. The probability of under-/over-voltage, line overloading, and so on can be judged. For instance, the upper limit of the real PF of branch 2–8 is supposed to be equal to 445 MW, i.e., corresponding to the vertical line in Figure 2, and the probability that power transmitted through the branch is over its limit can be computed as:

$$\mathbf{P}\{P_{2-8} > 445\} = 1.4\% \tag{7}$$

Similarly, the probability that voltage at a considered bus is out of the operating range can be assessed. However, in this test, the voltages at all buses in the system (for example, V_{89} in Figure 3) are within the range, i.e., [0.9, 1.1] p.u.

The results obtained by the ADCPPF method can help the operator of the system to evaluate the operating states in order to make suitable decisions and provide solutions for the system.

4. Further Discussion on Applicability of Various PPF Methods

PPF methods can be selected for application in both planning and operation problems of power systems.

- Applying PPF in planning problems: The MCS method is suitable for solving planning problems with long time frames (such as years, seasons, months, weeks) or operational planning problems in the time frame of a few days. In such cases, the time to achieve results does not need to be very fast. In addition to network configuration data, data on sources (especially RES) and loads collected over long periods, i.e., months, a year, or several years, are used to estimate PDFs. These data can also be used by a forecasting technique to provide results for operating the system. If the forecasting technique follows the point forecast approach, the forecast results are provided as a set value at each forecast time point and a corresponding error. These values are considered the expectation and standard deviation of the normal distribution function.

These functions are the input information of the MCS problem. If the forecasting technique follows an uncertainty forecast approach such as the probabilistic forecast or scenario forecast approaches [24], then the probability distribution function will be more useful. In fact, when applying MCS in practice, if the system is too large, with many input variables, the processing of MCS is very difficult, taking up an extremely large amount of memory, making it very difficult to process, and, in many cases, it may not be possible. In such cases, clustering (as in the current paper) and dimensionality reduction techniques should be used.

- Application in operational problems: For operational problems with extremely short time frames ranging from a few minutes (i.e., very short-term frame) to a few hours (i.e., short-term frame) and within 24 h (day-ahead), analytical methods and approximation methods with a fast processing time can be applied. The MCS method can also be used in the time frame of several hours or more when the power system is small in scale and the number of input RVs is also small. The MCS method combined with clustering techniques, such as the one proposed in this paper, can be implemented with short or even extremely short operating time frames. Based on data collected from RESs or loads (in addition to other network configuration data), probability distribution functions at the time points of the operational problem are built. These functions can also be provided by a forecast technique.

The next part is an illustrative example. Suppose a technique for forecasting the load or power generation of RESs provides forecast information at different time points in the operational time frame. At that time, the results obtained from PPF methods will be very useful for the system operator.

Figure 4 illustrates the PDF of the current flowing on a line of interest at consecutive times (resolution of 1 h) in the 24 h time horizon of day-ahead operation of the system. The system operator will have a very clear "picture" of the risk of system insecurity so that they can propose appropriate solutions. On the basis of comparing the PDF curve with the maximum limit Imax (corresponding to the limit of the power) of the line of interest, it is possible to determine the point at which there is a risk of the current passing through that line exceeding the allowable value.

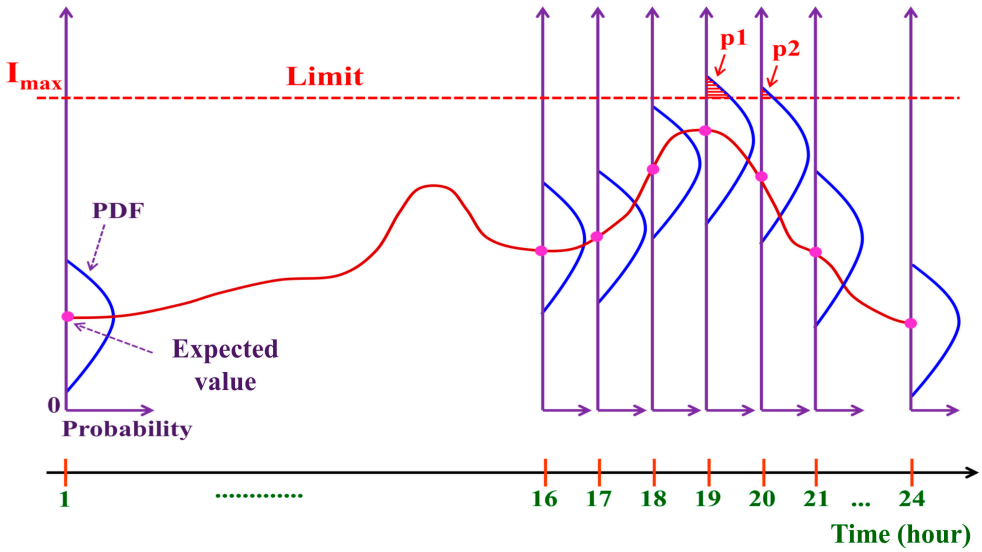


Figure 4. The example illustrates the PDF of current flowing on an interested line at consecutive times in the 24 h time horizon of day-ahead operation of the system.

For example, in Figure 4, from 16:00 to 18:00, the current gradually increases and, near 19:00, the upper boundary of the PDF line begins to touch and surpass the limit Imax and the calculated probability that corresponds to 19:00 is p1. It should be noted that, as the

evening approaches, the solar power source gradually decreases, and in this example, by 19:00, this source no longer generates power. However, this is the peak period in the system where the load increases rapidly until reaching the peak. At this time, the uncertainty in the system comes from the loads and other sources, if any, and not from the solar power source. Then, at 20:00, the current tends to decrease and the level of intrusion has a value of p2. Thus, in the above case, it can be roughly considered that the risk of overloading lasts about 1 h. The overload time can be estimated more accurately if the calculation time-step is smaller (30 min, 15 min, etc.), and this depends on the capabilities of the forecasting technique as well as the requirements of the system operator. If the period of risk of overloading lasts long but is still within the allowable limit, then the overload is considered a temporary overload and does not require any intervention. On the contrary, when there is a risk of overloading for a long period of time and the level of overload is severe, the operator must come up with suitable solutions to ensure security for the system.

It should be noted that, in Figure 4, if we only care about the expected value (similar to the results obtained from the traditional DPF), then all calculated current values are smaller than the limit Imax and safety risks due to the overloading are "not seen". This is a very new point in the view of system security assessment.

Figure 5 is an illustrative example of assessing the risk of over-/under-voltage at a bus of interest. In this example, the risk of under-voltage at times 19:00 and 20:00 is calculated as p1 and p2, respectively.

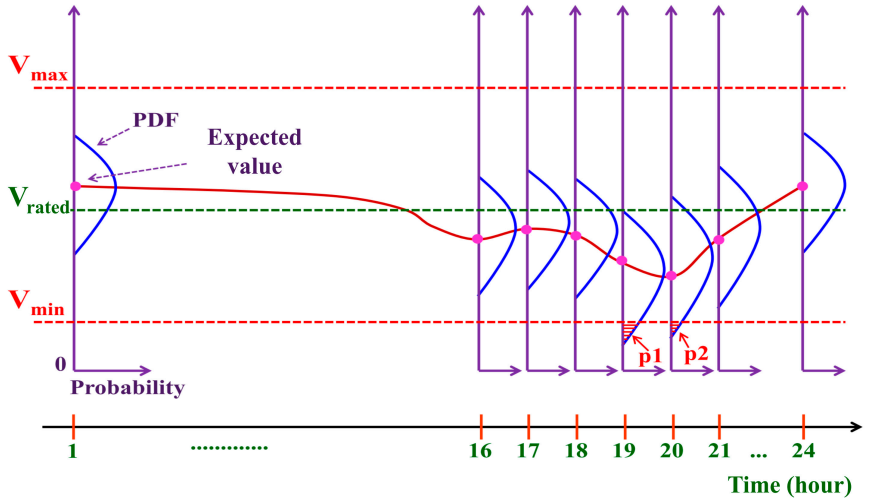


Figure 5. The example illustrates the PDF of the voltage at a node of interest at consecutive times in the 24 h time horizon of day-ahead operation of the power system.

The method of calculating and analyzing the security of the system is based on information and data obtained from random input quantities of the problem such as the loads and output powers of RESs. The result is in terms of probability distributions of output quantities such as the node voltage, or the power or current transmitted on branches. This process is performed before the real operation to find the probability distributions of the quantities of interest.

For actual power systems with a SCADA EMS system, this system will provide information about mode parameters and be updated regularly in close to real-time, so monitoring of the operation of mode parameters degrees is performed continuously. In addition, when there is a SCADA EMS system, the data collected for random factors will be more convenient and continuously updated. When the dataset of random factors is more complete, the information obtained is clearer. That is also another benefit when using a SCADA EMS system combined with the PPF method.

The main goal of this paper is to develop an approach to calculating PPF that ensures a certain acceptable accuracy but that also provides very fast results to meet the desired

application in the time-frame of almost "real time" in operation of the power system. As mentioned above, the MCS method faces many challenges and is even impossible when applied to real power systems, especially large-scale systems with very short operating time-frames. The clustering algorithm proposed to be applied for solving the PPF problem in this paper is simple, easy to implement, and helps effectively deal with the poor scalability of the K-means algorithm. Therefore, ADCPPF can give quick results with large data that can be applied to solve PPF problems for large-scale power systems.

However, in addition to the advantages and contributions of the proposed method in practical applications, it also has a limitation that needs to be addressed: predetermining the value of k like in the K-means algorithm. In addition, the inherent limitations of the PPF, which does not cover electromagnetic transients, and extreme abnormal operating conditions are unresolved issues. Power system planning and operation should also focus on worst case future and abnormal scenarios (e.g., heat waves, winter storms, etc.). The extreme weather events exacerbated by climate change and global warming pose a high level of uncertainty. Therefore, more diversity in types of uncertainty should be considered for integration into the PPF problem. Energy storage systems (e.g., batteries, pumped storage hydropower, and electric vehicle-to-grid schemes) can mitigate the uncertainty related to RESs that is also not considered in this study. Hardware is one of the important factors affecting PPF analysis, especially in relation to the calculation time. Future research can also focus on processing the algorithm in high-performance computers, overcoming the time processing concerns, and allowing more focus on the model accuracy. These limitations open up topics for consideration in future research.

5. Conclusions

PPF is an effective tool in calculating and analyzing power systems and considering uncertainties existing in the system. It can help the operator of the system in assessing the security. Among the various techniques for PPF, MCS gives highly accurate results but is often very computationally intensive. This article focuses on solving the problem of reducing the computation time for Monte Carlo simulation to achieve a practical tool with high accuracy that gives fast calculation results to be used to solve problems in the time horizon of power system operations. To achieve that goal, we make use of an advanced data clustering technique called K-means-lite. The developed approach, ADCPPF, is extensively tested on a modified IEEE-300 bus system, showing good performance in reducing computation time.

Author Contributions: Methodology, N.T.A.N., D.D.L., V.D.N., V.K.P. and V.K.H.; software, N.T.A.N. and D.D.L.; All authors wrote and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Education and Training, Vietnam under project number CT2022.07.DNA.03.

Data Availability Statement: Data are contained within the article.

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

References

1. Le, D.D.; Berizzi, A.; Bovo, C. A probabilistic security assessment approach to power systems with integrated wind resources. *Renew. Energy* **2016**, *85*, 114–123. [CrossRef]
2. Borkowska, B. Probabilistic load flow. *IEEE Trans. Power Appl. Syst.* **1974**, *PAS-93*, 752–759. [CrossRef]
3. Allan, R.N.; Al-Shakarchi, M.R.G. Probabilistic techniques in A.C. load-flow analysis. *Proc. Inst. Elect. Eng.* **1977**, *124*, 154–160. [CrossRef]
4. Zhang, P.; Lee, S.T. Probabilistic load flow computation using the method of combined cumulants and Gram-Charlier expansion. *IEEE Trans. Power Syst.* **2004**, *19*, 676–682. [CrossRef]
5. Fan, M.; Vittal, V.; Heydt, G.T.; Ayyanar, R. Probabilistic Power Flow Studies for Transmission Systems with Photovoltaic Generation Using Cumulants. *IEEE Trans. Power Syst.* **2012**, *27*, 2251–2261. [CrossRef]

6. Le, D.D.; Berizzi, A.; Bovo, C.; Ciapessoni, E.; Cirio, D.; Pitto, A.; Gross, G. A probabilistic approach to power system security assessment under uncertainty. In Proceedings of the Bulk Power System Dynamics and Control—IX Optimization, Security and Control of the Emerging Power Grid (2013 IREP Symposium), Crete, Greece, 25–30 August 2013.
7. Su, C.L. Probabilistic load-flow computation using point estimate method. *IEEE Trans. Power Syst.* **2005**, *20*, 1843–1851. [CrossRef]
8. Mohammadi, M.; Shayegani, A.; Adaminejad, H. A new approach of point estimate method for probabilistic load flow. *Int. J. Electr. Power* **2013**, *51*, 54–60. [CrossRef]
9. Liu, Y.; Gao, S.; Cui, H.; Yu, L. Probabilistic load flow considering correlations of input variables following arbitrary distributions. *Electr. Power Syst. Res.* **2016**, *140*, 354–362. [CrossRef]
10. Yu, H.; Chung, C.Y.; Wong, K.P.; Lee, H.W.; Zhang, J.H. Probabilistic Load Flow Evaluation with Hybrid Latin Hypercube Sampling and Cholesky Decomposition. *IEEE Trans. Power Syst.* **2009**, *24*, 661–667. [CrossRef]
11. Zou, B.; Xiao, Q. Solving Probabilistic Optimal Power Flow Problem Using Quasi Monte Carlo Method and Ninth-Order Polynomial Normal Transformation. *IEEE Trans. Power Syst.* **2014**, *29*, 300–306. [CrossRef]
12. Hajian, M.; Rosehart, W.D.; Zareipour, H. Probabilistic power flow by Monte Carlo simulation with Latin Supercube sampling. *IEEE Trans. Power Syst.* **2013**, *28*, 1550–1559. [CrossRef]
13. Leite da Silva, A.M.; Milhorance de Castro, A. Risk Assessment in Probabilistic Load Flow via Monte Carlo Simulation and Cross-Entropy Method. *IEEE Trans. Power Syst.* **2018**, *14*, 1193–1202. [CrossRef]
14. Carpinelli, G.; Caramia, P.; Varilone, P. Multi-linear Monte Carlo simulation method for probabilistic load flow of distribution systems with wind and photovoltaic generation systems. *Renew. Energy* **2015**, *76*, 283–295. [CrossRef]
15. Hagh, M.T.; Amiyan, P.; Galvani, S.; Valizadeh, N. Probabilistic load flow using the particle swarm optimisation clustering method. *IET Gener. Transm. Distrib.* **2018**, *12*, 780–789. [CrossRef]
16. Khalghani, M.R.; Ramezani, M.; Mashhadi, M.R. Probabilistic Power Flow Based on Monte-Carlo Simulation and Data Clustering to Analyze Large-Scale Power System in Including Wind Farm. In Proceedings of the 2020 IEEE Kansas Power and Energy Conference (KPEC), Manhattan, KS, USA, 13–14 July 2020.
17. Hashish, M.S.; Hasanien, H.M.; Ji, H.; Alkuhayli, A.; Alharbi, M.; Akmaral, T.; Turky, R.A.; Jurado, F.; Badr, A.O. Monte Carlo Simulation and a Clustering Technique for Solving the Probabilistic Optimal Power Flow Problem for Hybrid Renewable Energy Systems. *Sustainability* **2023**, *15*, 783. [CrossRef]
18. Zhang, R.; Chen, Z.; Li, Z.; Jiang, T.; Li, X. Two-stage robust operation of electricity-gas-heat integrated multi-energy microgrids considering heterogeneous uncertainties. *Appl. Energy* **2024**, *371*, 123690. [CrossRef]
19. Jiang, Y.; Ren, Z.; Li, W. Committed Carbon Emission Operation Region for Integrated Energy Systems: Concepts and Analyses. *IEEE Trans. Sustain. Energy* **2024**, *15*, 1194–1209. [CrossRef]
20. Li, Z.; Xu, Z.; Wang, P.; Xiao, G. Restoration of a Multi-Energy Distribution System with Joint District Network Reconfiguration via Distributed Stochastic Programming. *IEEE Trans. Smart Grid* **2024**, *15*, 2667–2680. [CrossRef]
21. Miraftabzadeh, S.M.; Colombo, C.G.; Longo, M.; Foiadelli, F. K-Means and Alternative Clustering Methods in Modern Power Systems. *IEEE Access* **2023**, *11*, 119596–119633. [CrossRef]
22. Olukanmi, P.O.; Nelwamondo, F.; Marwala, T. k-Means-Lite: Real Time Clustering for Large Datasets. In Proceedings of the 5th International Conference on Soft Computing and Machine Intelligence, Nairobi, Kenya, 21–22 November 2018; pp. 54–59.
23. Wang, X.; Chiang, H.D.; Wang, J.; Liu, H.; Wang, T. Long-term stability analysis of power systems with wind power based on stochastic differential equations: Model development and foundations. *IEEE Trans. Sustain. Energy* **2015**, *6*, 1534–1542. [CrossRef]
24. Le, D.D.; Gross, G.; Berizzi, A. Probabilistic modeling of multisite wind farm production for scenario-based applications. *IEEE Trans. Sustain. Energy* **2015**, *6*, 748–758. [CrossRef]
25. Salameh, Z.M.; Borowy, B.S.; Amin, A.R.A. Photovoltaic module-site matching based on the capacity factors. *IEEE Trans. Energy Convers.* **1995**, *10*, 326–332. [CrossRef]
26. Power System Test Case Archive. Available online: http://labs.ece.uw.edu/pstca/pf300/pg_tca300bus.htm (accessed on 6 June 2024).

electronics

MDPI

Article

A Fast Inverse Synthetic Aperture Radar Imaging Scheme Combining GPU-Accelerated Shooting and Bouncing Ray and Back Projection Algorithm under Wide Bandwidths and Angles

Jiongming Chen [1], Pengju Yang [1,2,*], Rong Zhang [1] and Rui Wu [1,3]

[1] School of Physics and Electronic Information, Yan'an University, Yan'an 716000, China; jmchen@yau.edu.cn (J.C.); zr2108700733@yau.edu.cn (R.Z.); wurui@yau.edu.cn (R.W.)
[2] Key Laboratory for Information Science of Electromagnetic Waves (MoE), Fudan University, Shanghai 200433, China
[3] Institute for Radio Frequency Technology and Software, Beijing Institute of Technology, Beijing 100081, China
* Correspondence: pjyang@fudan.edu.cn

Abstract: Inverse synthetic aperture radar (ISAR) imaging techniques are frequently used in target classification and recognition applications, due to its capability to produce high-resolution images for moving targets. In order to meet the demand of ISAR imaging for electromagnetic calculation with high efficiency and accuracy, a novel accelerated shooting and bouncing ray (SBR) method is presented by combining a Graphics Processing Unit (GPU) and Bounding Volume Hierarchies (BVH) tree structure. To overcome the problem of unfocused images by a Fourier-based ISAR procedure under wide-angle and wide-bandwidth conditions, an efficient parallel back projection (BP) imaging algorithm is developed by utilizing the GPU acceleration technique. The presented GPU-accelerated SBR is validated by comparison with the RL-GO method in commercial software FEKO v2020. For ISAR images, it is clearly indicated that strong scattering centers as well as target profiles can be observed under large observation azimuth angles, $\Delta\varphi = 90^{\circ}$, and wide bandwidths, 3 GHz. It is also indicated that ISAR imaging is heavily sensitive to observation angles. In addition, obvious sidelobes can be observed, due to the phase history of the electromagnetic wave being distorted resulting from multipole scattering. Simulation results confirm the feasibility and efficiency of our scheme by combining GPU-accelerated SBR with the BP algorithm for fast ISAR imaging simulation under wide-angle and wide-bandwidth conditions.

Keywords: ISAR imaging; shooting and bouncing ray method; GPU acceleration; BVH tree; back projection algorithm

Citation: Chen, J.; Yang, P.; Zhang, R.; Wu, R. A Fast Inverse Synthetic Aperture Radar Imaging Scheme Combining GPU-Accelerated Shooting and Bouncing Ray and Back Projection Algorithm under Wide Bandwidths and Angles. *Electronics* **2024**, *13*, 3062. https://doi.org/10.3390/electronics13153062

Academic Editor: Chiman Kwan

Received: 4 June 2024
Revised: 29 July 2024
Accepted: 30 July 2024
Published: 2 August 2024

1. Introduction

Inverse synthetic aperture radar (ISAR) is a powerful active microwave imaging radar system widely utilized in military and civil applications due to its capability to produce high-resolution images for moving targets in almost all-weather and all-day conditions [1–3]. ISAR images can be obtained by focusing scattering field data at multiple angles and frequencies, which are a two-dimensional representation of the target scattering center [4–6]. The simulation of ISAR imaging for electrical large targets is extremely time-consuming, due to the calculation of multiple angles and frequencies' scattering field. Several methods and their improved versions as well as acceleration techniques have been developed to efficiently calculate the scattering from electrical large targets, including both the low-frequency numerical method and high-frequency approximation methods [7,8]. Due to the tremendous computational time and memory requirements, pure numerical methods such as method of moments (MoM) and finite element method (FEM) are facing enormous challenges [9]. Due to the good compromise between accuracy and efficiency, high-frequency approximation methods are widely utilized in ISAR imaging simulation for

electrical large targets. Among them, the shooting and bouncing ray (SBR) method is the most popular one, which is a combination of physical optics (PO) and geometrical optics (GO) and is suitable for taking multiple scattering into account.

Following the proposal of the SBR method by Ling in 1989, researchers have implemented numerous improvements [10], such as the time domain shooting and bouncing ray method (TDSBR), bidirectional analytic ray tracing [11], etc. These advancements have contributed to the growing popularity of the bouncing ray method. In recent years, a GPU-accelerated bouncing ray method is proposed based on a stackless k-dimension (Kd) tree traversal algorithm, enabling the ray tracing process to be efficiently carried out in the GPU [12]. In [13], an enhanced bouncing ray method using a ray propulsion technique is proposed to accelerate the ray tracing process and enhance the ray intersection efficiency, making it able to efficiently calculate the scattering characteristics of electrically large targets. In [14], the inclusion of reverse ray paths in the SBR method is proposed to improve the accuracy of cavity radar cross section (RCS) predictions, which can be implemented into existing SBR code almost trivially, while producing potentially substantial improvements in prediction accuracy. A reverse ray tracing technique is proposed based on the ropes Kd-tree data structure, which has been demonstrated to yield satisfactory results in the calculation of high-frequency scattering characteristics [15]. In addition to the enhancement of SBR through the utilization of GPUs and data structures, the SBR method has also been integrated with other electromagnetic computational methods, thereby rendering this method more comprehensive in its consideration of the electromagnetic scattering characteristics of complex target structures [16–19]. For instance, the octree-based SBR method in combination with the physical theory of diffraction (PTD) is presented for the analysis of electromagnetic (EM) scattering from the moving target [20]. In [21], a hybrid method of equivalent dipole moment (EDM), MOM, and SBR is proposed to enhance the computational efficiency of the RCS of complex objects within the EDM framework. In this hybrid method, an iterative approach is introduced to enhance the algorithm performance, offering high accuracy and reducing the computational time.

On the basis of electromagnetic scattering modeling, focused ISAR images can be obtained by applying a signal processing algorithm, including the range Doppler (RD) algorithm, polar format algorithm (PFA), back projection (BP) algorithm, etc. [22–24]. The most commonly used ISAR imaging algorithm interpolates the polar data to a Cartesian grid and then applies a 2-D FFT to achieve ISAR reconstruction. As a special case, under small-angle and small-bandwidth conditions, ISAR images can be approximately obtained by performing inverse Fourier transform of 2D backscattered field data, and the resultant ISAR images are composed of the scattering centers of target with their electromagnetic reflection coefficient. Due to its suitability for GPU parallel processing and ability for ISAR imaging in any mode, the BP algorithm and its modified versions are extensively used in SAR/ISAR imaging applications [25]. As early as in 1989, a simplified BP algorithm and its parallel processing architecture were proposed by using the radar waveform as the impulse response of the filter to obtain the filtered projection [26]. Up to now, the GPU-based BP algorithm is still being developed to optimize the peak performance of the BP algorithm on servers and miniaturized GPU devices, which can deal with the differences in hardware platforms as well as the differences in data scales [27]. In [28], an ISAR imaging algorithm for composite target–ocean scenes based on time-domain shooting and bouncing rays (TDSBRs) is developed. In [29], time-domain iterative physical optics (TD-LIPO) is proposed to analyze scattering from electrically large and complex targets. In [30], an accelerated time-domain iterative physical optics method is developed for analyzing the scattering from electrically large and complex targets, and an IFFT is performed to obtain the ISAR image under small-bandwidth and small-angle conditions.

Aiming at ISAR imaging for electrical large targets under large-bandwidth and wide-angle conditions, this paper is devoted to a scheme of ISAR imaging by combining GPU-accelerated SBR based on GPU and BVH tree acceleration with the GPU-accelerated BP algorithm. To enhance ray intersection efficiency, a BVH tree structure is constructed

according to the target structure, which is implemented in C++AMP to achieve GPU parallel acceleration computation. The SAH method is incorporated into the scene bounding box division, effectively mitigating the impact of bounding box overlapping on the ray traversal efficiency of the BVH tree structure. To efficiently perform ISAR imaging simulation, a GPU-based accelerated BP imaging algorithm is developed by virtue of a compute unified device architecture (CUDA).

This paper is organized as follows: Section 2 introduces a GPU-accelerated SBR using BVH tree structure. In Section 3, the GPU-accelerated BP imaging algorithm is presented. In Section 4, the results and discussion are presented, and several simulations are performed to confirm the feasibility and efficiency of our scheme by combining GPU-accelerated SBR with the BP algorithm for fast ISAR imaging simulation under wide-angle and wide-bandwidth conditions. Section 5 concludes this paper.

2. A GPU-Accelerated SBR Using BVH Tree Structure

2.1. Calculation of Multiple Scattering Using PO and GO

As a high-frequency approximation method, the shooting and bouncing ray method is a combination of PO and GO, which uses GO to trace the electromagnetic wave reflection path and PO to calculate the scattering field, resulting in a great advantage in solving electromagnetic scattering problems for complex targets [31]. According to PO approximate theory, target surfaces are divided into bright and dark areas, depending on whether the surfaces are illuminated by electromagnetic waves or not. The total field scattered from the dark area is assumed to be zero. However, this assumption is only valid when the wavelength of the electromagnetic wave is much smaller than the target geometry. It becomes difficult to determine the total field of the area on the target that is not directly illuminated by the incident wave when the target's dimensions are very large along the perpendicular direction of the incident wave. Assuming a total field of zero on the shaded surface would imply a discontinuity in the field on the shaded boundary. Therefore, to resolve the discontinuity in the boundary field, a line integral must be added on the boundary [32,33]. In Figure 1, a schematic diagram of multiple scattering of SBR is illustrated, in which only the 1st reflection (in yellow arrows) and 2nd reflection (in green arrows)are depicted.

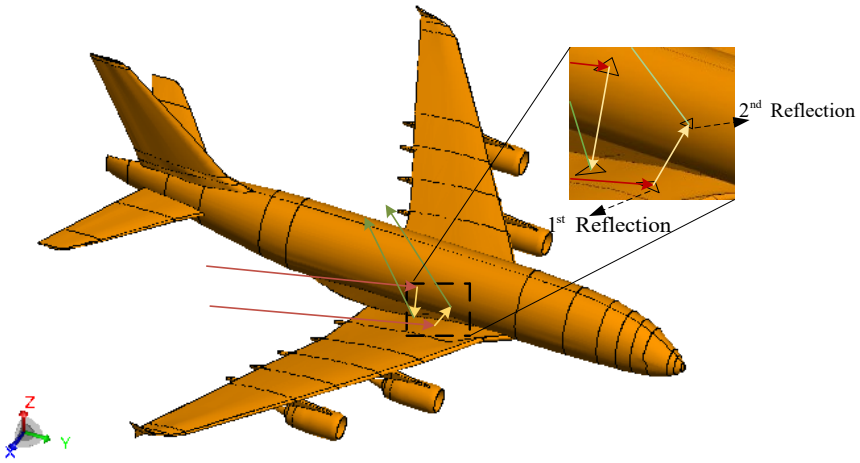


Figure 1. Schematic diagram of the multiple scattering.

The PO field of the perfect electric conductor (PEC) target at position $\boldsymbol{r}_s$ can be expressed as [34]

$$\begin{aligned} \boldsymbol{E}_s^{po}(\boldsymbol{r}_s) &= -jk\eta \int_S \left[\hat{\boldsymbol{k}}_s \times \left(\hat{\boldsymbol{k}}_s \times \boldsymbol{J}\right)\right] G(\boldsymbol{r}_s, \boldsymbol{r}') dS' \\ &= -jk\eta \frac{H_i \mathrm{e}^{-jk(r_s - \boldsymbol{r}')}}{2\pi r_s} \int_S \left\{\hat{\boldsymbol{k}}_s \times \left[\hat{\boldsymbol{k}}_s \times \left(\hat{\boldsymbol{n}} \times \hat{\boldsymbol{h}}_i\right)\right]\right\} \mathrm{e}^{-jk(\hat{\boldsymbol{k}}_i - \hat{\boldsymbol{k}}_s)\cdot \boldsymbol{r}'} dS' \end{aligned} \quad (1)$$

where $\hat{\boldsymbol{h}}_i$ is direction vector of the incident magnetic field and $\hat{\boldsymbol{k}}_i$ is electromagnetic wave propagation direction vector. $\hat{\boldsymbol{k}}_s$ is the unit vector of the scattering wave direction. S represents the area illuminated by the incident wave, and $\boldsymbol{r}'$ is the source point location. In far field zone, Green's function in free space can be approximated as [35]

$$\mathrm{G}(\boldsymbol{r},\boldsymbol{r}') \approx \frac{\mathrm{e}^{-jk\cdot(\boldsymbol{r}-\boldsymbol{r}')}}{4\pi|\boldsymbol{r}|} \tag{2}$$

When the radius of curvature at a point on the target is much larger than the wavelength of the incident wave, tangential plane approximation can be applied, and the induced current can be expressed as [36]

$$\boldsymbol{J} = \hat{\boldsymbol{n}} \times \boldsymbol{H} = \begin{cases} 2\hat{\boldsymbol{n}} \times \boldsymbol{H}_i & \text{Illuminated region} \\ 0 & \text{Shadow region} \end{cases} \tag{3}$$

In Figure 2, D_m is the area of the ray tube at the m ray, and D_{m+1} is the area of the ray tube at the reflected ray of the m ray. Rays cause changes in the amplitude and phase of the electric field as they propagate and reflect, and information about the strength and phase of the electric field of each ray tube is tracked at each reflection point. In GO, the electric field at $\boldsymbol{r}_m$ is related to the electric field at its reflected $\boldsymbol{r}_{m+1}$ according to the following equation.

$$\boldsymbol{E}^s(\boldsymbol{r}_{m+1}) = \Gamma_m \cdot \boldsymbol{E}^s(\boldsymbol{r}_m) \cdot (DF)_m \cdot \mathrm{e}^{-j\beta} \tag{4}$$

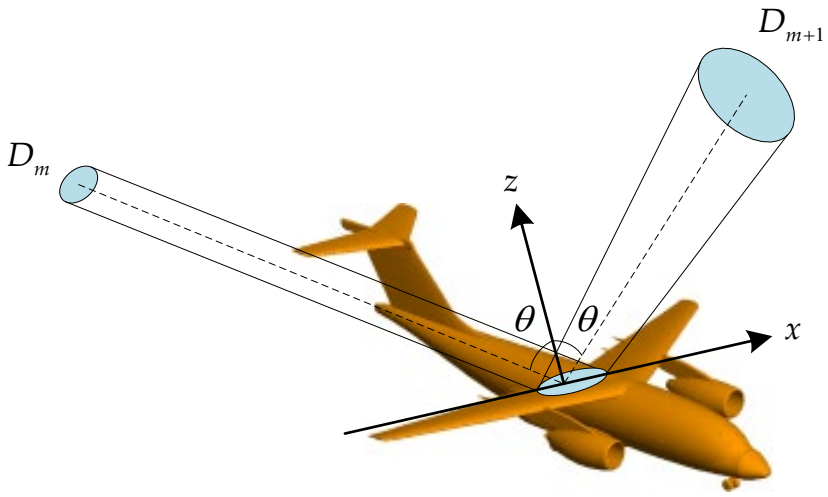


Figure 2. Schematic diagram of beam reflectance. The shape of the ray tube changes after each reflection, which is determined by the divergence factor $(DF)_m$.

In Equation (4), Γ_m denotes the reflection coefficient at $\boldsymbol{r}_m$. For a perfect conductor, $\Gamma_m = -1$ for horizontal polarization and $\Gamma_m = 1$ for vertical polarization. The divergence factor $(DF)_m$ at $\boldsymbol{r}_m$ is usually denoted by $(DF)_m \approx \sqrt{D_m/D_{m+1}}$.$\beta = k_0|\boldsymbol{r}_{m+1} - \boldsymbol{r}_m|$, which represents the phase difference between two neighboring reflection points. Substituting Equation (1) into Equation (4), the scattered field after reflection of the m-*th* ray can be expressed as [37–40].

$$\begin{aligned} \sum_x \boldsymbol{E}^{po}_{\Delta x}(\boldsymbol{r}_m) &= -jk\eta\Gamma_m(DF)_m \mathrm{e}^{-jk|\boldsymbol{r}_{m+1}-\boldsymbol{r}_m|} \sum_x \int_{\Delta x} \left[\hat{\boldsymbol{k}}_s \times (\hat{\boldsymbol{k}}_s \times \boldsymbol{J})\right] G(\boldsymbol{r}_m,\boldsymbol{r}')dS' \\ &= -jk\eta\Gamma_m(DF)_m \mathrm{e}^{-jk|\boldsymbol{r}_{m+1}-\boldsymbol{r}_m|} \frac{H_i \mathrm{e}^{-jk(\boldsymbol{r}_m-\boldsymbol{r}')}}{2\pi|r|} \sum_x \int_{\Delta x} \left\{\hat{\boldsymbol{k}}_s \times \left[\hat{\boldsymbol{k}}_s \times (\hat{\boldsymbol{n}} \times \hat{\boldsymbol{h}}_i)\right]\right\} \mathrm{e}^{-jk(\hat{\boldsymbol{k}}_i-\hat{\boldsymbol{k}}_s)\cdot\boldsymbol{r}'} dS' \end{aligned} \tag{5}$$

In Equation (5), $\sum_{x} \boldsymbol{E}_{\Delta x}^{po}(\boldsymbol{r}_m)$ is the PO field generated by the facet element struck by the m-*th* ray, and x is the number of facet elements struck. The resulting scattered field for each ray is subsequently superimposed to obtain the total scattered field.

$$\boldsymbol{E}_s^{SBRtotal}(r_m) = \sum_{1}^{l} \sum_{x} \boldsymbol{E}_{\Delta x}^{po}(\boldsymbol{r}_m) \tag{6}$$

In Equation (6), l is the total number of rays and Δx is the area of some triangular face element that was hit.

2.2. GPU-Accelerated Ray Tracing Using BVH Tree Structure

2.2.1. GPU Acceleration Process

In this paper, GPU-accelerated SBR is parallelized using C++ accelerated massive parallelism (C++AMP). In comparison with CPUs, GPUs possess a greater number of cores, making them more suitable for massive parallel processing. C++AMP is a C++-based heterogeneous parallel computing platform released by Microsoft, which is a native programming model with the advantage of running across devices on the Windows platform [41]. Most of the programming methods for GPUs, such as Direct Compute and OpenCL, require different programming languages and compilers. C++AMP unifies the programming language and the compiler, which sets it apart from other approaches. The C++ AMP library enables parallel computation through a set of abstractions and a high-level API, with the underlying GPU hardware being accessed directly through Direct Compute [42,43]. It is important to allocate an array for applying C++AMP to implement parallel computing. The array template is in the concurrency namespace. It takes two parameters: one for the collection element type and the other for the dimension. The dimension of the array is set according to the type of collection elements in this paper method. For example, when collecting swept frequency data, defining an <*array a* (*frequency counts*)>, this example defines a one-dimensional array with the size of the array being the number of frequencies. Arrays play an extremely important role in C++AMP by representing a view that can access data on the GPU and encapsulating C++ arrays or vectors, which are arrays on <*accelerator_view*>. In C++AMP, the GPU is not the only accelerator, and each accelerator has its own default view. Once the array has been constructed, the data will be transferred to GPU memory, where it can be accessed directly by the GPU. The <*parallel_for_each*> function is used to execute parallel computation tasks. The <*parallel_for_each*> function is a parallel execution function in C++AMP that accepts a range of indexes and a lambda function, and it executes this lambda function in parallel on the GPU for each index. The <*parallel_for_each*> function delegates parallel computing tasks to the GPU's kernel functions, which can be directly assigned to the GPU hardware through the Direct Compute API. The <*restrict*(*amp*)> keyword is used to specify that the function is to be executed only on the GPU. It is used to identify specific blocks of code and lambda functions to be executed on the GPU, and it lets the compiler optimize the function for Single Instruction, Multiple Threads (SIMT) [44]. The operations within the function will utilize SIMT instructions to achieve the effect of single instruction multi-thread parallel computation. C++AMP synchronizes the data and copies them from GPU memory back into host memory after the <*parallel_for_each*> function executes the parallel computation task.

Figure 3 gives a parallel computation process for C++AMP using the Direct Computing API to send parallel instructions to the GPU device. The BVH tree structure as well as the ray data, etc., are constructed in the CPU and stored in the Global Memory of the GPU. The data will be automatically copied from the CPU host to GPU memory by creating the <*array_view*> array, allowing the GPU to access these data directly. The Constant Memory in the GPU will store data that remain unchanged during parallel computation. Texture Memory is used to store data during model rendering. There are many Streaming Multiprocessors (SMs) in the GPU hardware architecture, and SMs in GPUs use the SIMT architecture. Each SM contains a number of streaming processors (SPs), and each SP

corresponds to a thread. A single warp is comprised of 32 threads, with the number of threads per warp being determined by the GPU architecture. Warp is the scheduling and execution unit in SM, where threads in each warp execute computations in parallel. As previously stated, the <*parallel_for_each*> function assigns tasks directly to the underlying GPU hardware through the Direct Compute API. The GPU assigns instructions to warps in the SM with the SIMT architecture. The SIMT architecture permits the threads within each warp to execute a sequence of instructions, including collision detection, ray tracing, PO field calculation, and other operations, in parallel modes. In the SIMD architecture, the GPU assigns a thread to each ray. The BVH tree structure is obtained by thread accessing the GPU's Global Memory and is responsible for calculating the ray tracing path and the PO field. In the <*parallel_for_each*> function, intersection detection and the parallel acceleration of rays and bounding boxes are achieved by traversing the BVH tree structure. After the execution of the <*parallel_for_each*> function is completed, C++AMP copies the scattered field data from GPU memory to host memory. The CPU will sum the scattered fields of all the rays to obtain the total target scattering field.

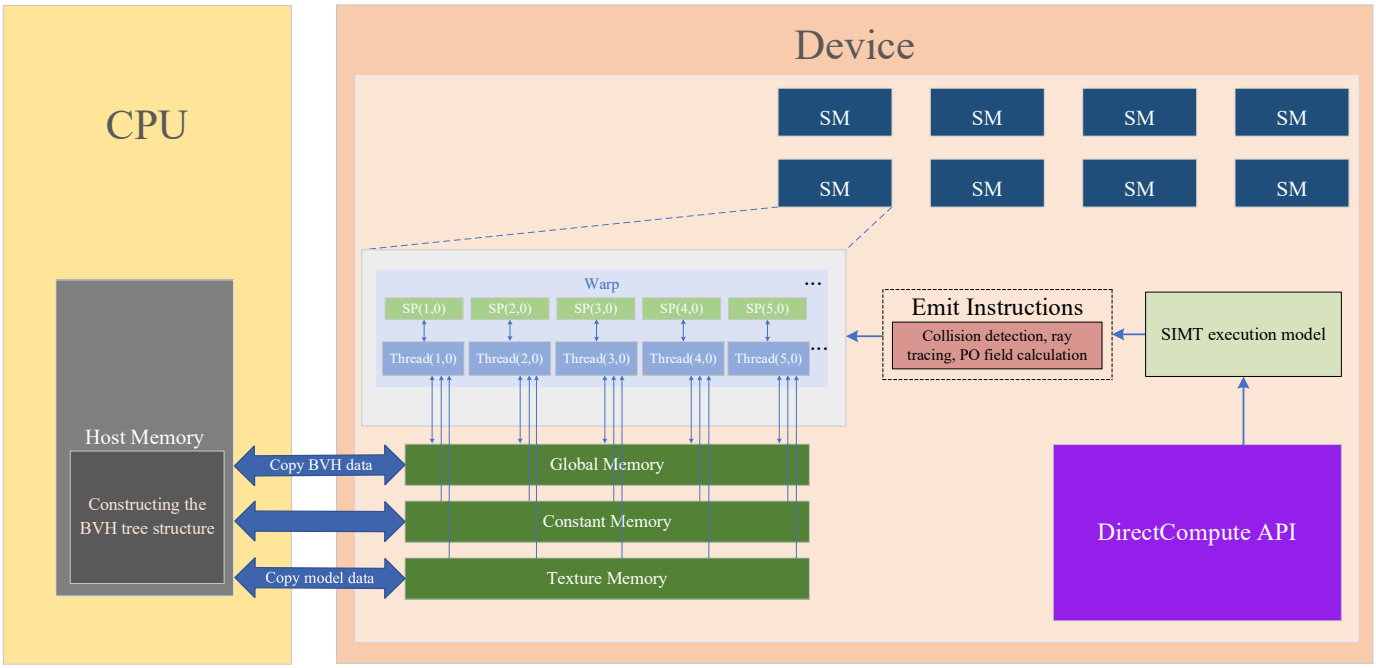


Figure 3. The parallel calculation process of C++AMP using the Direct Compute API to send parallel instructions to the device (GPU).

The ray information is initialized, and the nodes as well as the structure of the BVH tree are generated in the CPU as illustrated in Figure 4. The CPU will share the data of the BVH tree to the GPU Global Memory and it will share these data with each thread. In the SIMT architecture, instructions are emitted to the warp by the Direct Compute API. Threads in a warp will execute the received instructions sequentially and in parallel, and each thread in a warp will be responsible for the computation of one ray. At this stage, all the rays traverse the BVH tree intersecting the target are recorded, and their reflected rays are traced until the ray leaves the target surface. The GPU calculates the scattered field of this ray and transfers the data to the CPU. Finally, the CPU combines the scattered field data from all the rays to determine the total scattering field from the target.

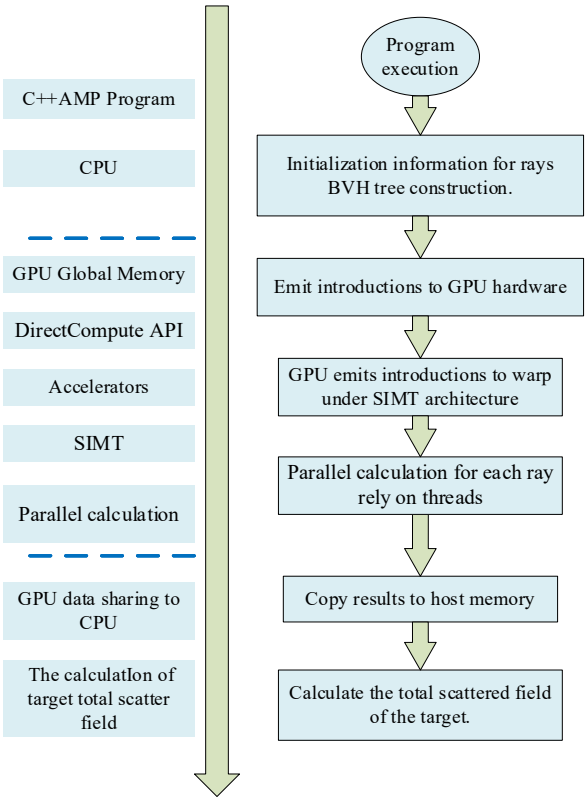


Figure 4. The structure of programming model for the GPU acceleration process.

2.2.2. Ray Tracing Algorithm Using BVH Tree

Figure 5 illustrates the ray tracing process using BVH tree structure with the multiple scattering process of an electromagnetic wave between the tree structures. GO is used to track the scattering path of the electromagnetic wave between the triangular surface tuples, and PO is used to calculate the scattered field of the electromagnetic wave when it hits the triangular surface tuples.

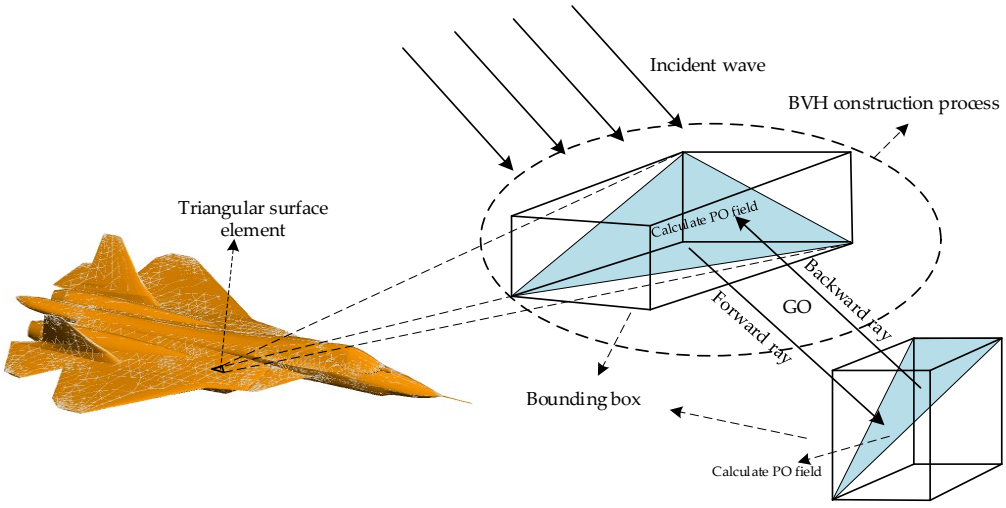


Figure 5. BVH tree structure accounting for multiple scattering between triangular surface patches.

In order to solve the time-consuming ray intersection process in ray tracing, we resort to BVH tree structure on the basis of GPU acceleration. A BVH tree is a computer graphics structure, which is a tuple-based ray intersection detection technique. A BVH tree divides the tuples into a hierarchical structure of disjoint sets, which is widely used in ray tracing

and collision detection. We will build the bounding box that encloses the target according to the target geometric features, and its enclosure box structure is an axisymmetric bounding box, an AABB bounding box. In the BVH tree structure, all the tuples are stored in the leaf nodes of the BVH tree, and the middle nodes store the box information. Finally, the whole scene's information is stored in the BVH tree structure [45]. When traversing the BVH tree, the ray will first judge whether it intersects with the box or not. If it is not intersected, then it will skip all the tuples in the box, achieving an improvement for the ray tracing efficiency.

In Figure 6, the scene contains eight tuples. In Step ①, the tuples are divided into two parts in the scene. If the light rays do not intersect with the bounding box, then they will not intersect with the tuples, which can exclude half of the tuples at one time and thus reduce the number of intersections. In Step ②, we continue to divide the tuples in half, and by doing so, the ray intersection complexity can be reduced from $O(n)$ to $O(\log(n))$ [46]. After pairwise semi-recursive division, the bounding box that surrounds only one tuple can be found. If a ray intersects this bounding box, the ray continues to intersect with the tuple. The ray parameter equation can be expressed as

$$\boldsymbol{r}(t) = \boldsymbol{o} + t \cdot \boldsymbol{d} \tag{7}$$

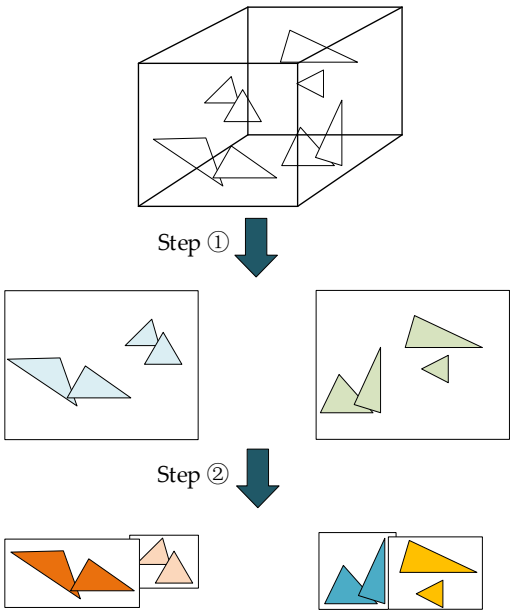


Figure 6. Segmentation process of scenario tuples.

In Equation (7), $\boldsymbol{o}$ and $\boldsymbol{d}$ are ray starting point and the normalized ray direction vector, respectively. Substituting the ray parameter equation into the implicit plane equation of the plane where the tuple is located, the implicit plane equation can be written as

$$\boldsymbol{N}^T(\boldsymbol{o} + t \cdot \boldsymbol{d}) = c \tag{8}$$

From Equation (8), the parameter t corresponding to the intersection point of the ray with the plane can be resolved as

$$t = \frac{c - \boldsymbol{N}^T \cdot \boldsymbol{o}}{\boldsymbol{N}^T \cdot \boldsymbol{d}} \tag{9}$$

In this paper, the surface element is a triangular surface element, and the plane parametric equation of the triangular surface element is

$$f(u, v) = (1 - u - v) \cdot \boldsymbol{p}_0 + u \cdot \boldsymbol{p}_1 + v \cdot \boldsymbol{p}_2 \tag{10}$$

where u and v are the triangle center of mass coordinates, satisfying $u \geq 0$, $v \geq 0$, and $u + v \leq 1$. The triangular face element can be regarded as the mapping of the unit triangle face element on its three edges

$$f(u, v) = \boldsymbol{p}_0 + u(\boldsymbol{p}_1 - \boldsymbol{p}_0) + v(\boldsymbol{p}_2 - \boldsymbol{p}_0) \tag{11}$$

Combining Equation (11) with Equation (7), one can obtain

$$\underbrace{[\boldsymbol{p}_1 - \boldsymbol{p}_0 \boldsymbol{p}_2 - \boldsymbol{p}_0 - \boldsymbol{d}]}_{M^{-1}} \begin{bmatrix} u \\ v \\ t \end{bmatrix} = \boldsymbol{o} - \boldsymbol{p}_0 \tag{12}$$

In Equation (12), M^{-1} is the matrix transforming a triangular face element into a unit triangular face element in the u, v plane, in which the mapped ray is orthogonal to the unit triangular face element. The mapping of triangular surface elements into the u, v plane is illustrated in Figure 7. Figure 7a is the case before mapping, and Figure 7b is the case corresponding to Equation (11). Figure 7c accounts for the intersection of the ray with the unit triangular surface element after mapping.

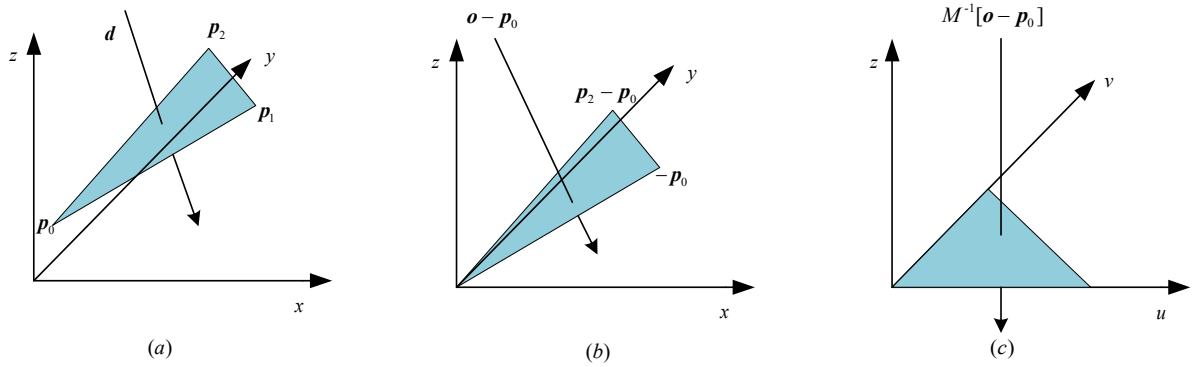


Figure 7. Mapping of triangular surface elements into u, v plane. (**a**) The case before mapping; (**b**) The case corresponding to Equation (11); (**c**) The intersection of the ray with the unit triangular surface element after mapping.

According to the half-division method, the bounding boxes in the scene may overlap or intersect, and the overlapping of bounding boxes is illustrated in Figure 8. In Figure 8, when the light traverses the BVH tree, the overlapping of bounding boxes causes the light to intersect both bounding boxes, which leads to a decrease in the intersection efficiency.

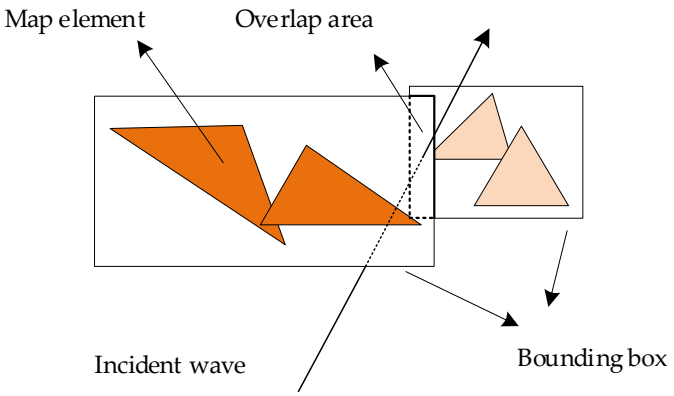


Figure 8. Overlap diagram of child nodes' bounding boxes.

In order to eliminate the phenomenon of envelope box overlapping, the SAH division method instead of the half-division method is adopted in the process of BVH tree construction. The SAH is based on the surface area heuristic division method, and after adding SAH, we can estimate the probability of the light ray hitting the enveloping boxes in terms of the size of the surface area of the parent enveloping box in which there are two or more overlapping child enveloping boxes.

In the BVH tree structure, under the assumption that the current node has three bounding boxes A, B, and C, the cost of intersecting the ray with the current node is

$$c(A,B,C) = p(A)\sum_{i\in A} t(i) + p(B)\sum_{i\in B} t(j) + p(C)\sum_{i\in C} t(k) + t_{trav} \quad (13)$$

In Equation (13), $\sum t(i)$ is the intersection cost of each sub-enclosure box, and $t(i)$ is the *i-th* tuple in the child enclosure box. $p(A)$, $p(B)$, and $p(C)$ are the probabilities of the light hitting the objects in the bounding boxes A, B, and C, respectively. t_{trav} is the cost of the light traversing the BVH tree.

In SAH, we use the surface area of the child bounding box in the parent node instead of the probability of the ray hitting the bounding box. Assuming that the surface areas of the child bounding boxes A, B, and C are $S(A)$, $S(B)$, and $S(C)$, respectively, and the surface of the parent node bounding box D is $S(D)$, Equation (13) can be rewritten as

$$c(A,B,C) = \frac{S(A)}{S(D)}\sum_{i\in A} t(i) + \frac{S(B)}{S(D)}\sum_{i\in B} t(j) + \frac{S(C)}{S(D)}\sum_{i\in C} t(k) + t_{trav} \quad (14)$$

After resolving the optimal division method by calculating the minimum value of Equation (14), the ray traversal of the BVH tree is most efficient [47].

3. GPU-Accelerated BP Imaging Algorithm

3.1. BP Algorithm for ISAR Imaging

Under small-angle and small-bandwidth conditions, ISAR images can be approximately obtained by performing the inverse Fourier transform of 2D backscattered field data, and the resultant ISAR images are composed of the scattering centers of the target with their electromagnetic reflection coefficient. Due to its suitability for GPU parallel processing and ability for ISAR imaging in any mode, the BP algorithm and its modified versions are extensively used in SAR/ISAR imaging applications. The BP imaging algorithm is a method with high imaging accuracy. However, due to its high complexity, the BP algorithm is not as good as other imaging algorithms in terms of imaging speed. Aiming at ISAR imaging for electrical large targets under large-bandwidth and wide-angle conditions, a GPU-based accelerated BP imaging algorithm is developed in this paper by virtue of CUDA, while maintaining the imaging accuracy of the BP algorithm.

The fundamental idea of the BP algorithm involves coherently superimposing the calculated echoes of each pulse by transmitting electromagnetic pulses and calculating the two-way time delay between the pixel points in the imaging area and the radar at the moment of each pulse. The superimposition depends on the phase relationship between pixel points. If the echoes are in phase, the superimposed pixel points' echoes become increasingly stronger. When pixel points with different phases are superimposed, the effect is weaker. As an accurate time-domain algorithm, the range profile in the BP algorithm is obtained using the pulse compression technique, similar to the Range-Doppler algorithm. The processing of the azimuthal direction is achieved by computing the echoes of the pixel points for coherent superposition, which is related to the angle of rotation of the target with respect to the radar. Azimuthal resolution is observed to increase as the angle between the target and the radar increases, with no apparent limit. For any motion trajectory of the target, if the motion trajectory can be predicted in advance, then the BP algorithm can achieve accurate imaging.

Figure 9 illustrates the schematic diagram of backward projection for ISAR imaging. xyz is a global coordinate system of space target. XY is the local coordinate system. S is the grid for the imaging area. R_a and R_b denote the radar distance from the target at moments a and b, respectively. The imaging area is divided into $N \times N$ grids. (X_a, Y_a) and (X_b, Y_b) are the positions of the target in the imaging grid at time a and b, respectively. v is the speed and direction of the moving target. The transmitting signal is [48]

$$s(t,m) = \text{rect}\left(\frac{t}{T_p}\right) \cdot \exp\left(j2\pi f_0 t + jK\pi t^2\right) \tag{15}$$

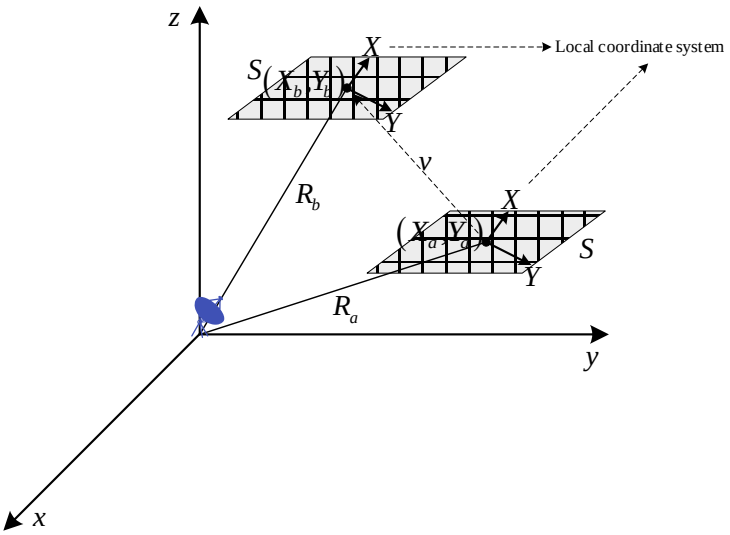


Figure 9. Schematic diagram of backward projection for ISAR imaging.

In Equation (15), t is fast time. T_p is the signal pulse width. m is slow time. rect is a rectangular window function. f_0 is the signal carrier frequency. $K = \frac{B}{T_p}$, K is the signal modulation frequency. B is the signal bandwidth. The received echo is as follows

$$s_r(t,m) = \sum_{i,j} \text{rect}\left(\frac{t - \tau_{i,j}(m)}{T_p}\right) \exp\left(j2\pi f_0\left(t - \tau_{i,j}(m)\right) + j\pi K\left(t - \tau_{i,j}(m)\right)^2\right) \tag{16}$$

In Equation (16), $\tau_{i,j}(m) = \frac{2\sqrt{(x(m)+X_i)^2+(y(m)+Y_i)^2+z(m)^2}}{c}$ is the two-way delay from pixel (X_i, Y_i) in the imaging plane to the radar at the slow time m. $x(m)$, $y(m)$, and $z(m)$ are the positions of pixel (X_i, Y_i) in the imaging grid in the spatial target coordinate system. The matched filter is as follows

$$h(t,m) = \exp\left(j2\pi f_0 t - jK\pi t^2\right) \tag{17}$$

In Equation (17), t_0 is the two-way delay at the closest distance between the target and the radar. Converting the time domain convolution to frequency domain multiplication processing, the matched filter output can be expressed as follows

$$s_{out}(t,m) = \text{IFFT}(\text{FFT}(s_r(t,m)) \cdot \text{FFT}(h(t,m))) \tag{18}$$

The high resolution in azimuthal is obtained by the coherent accumulation of pulses. The integral formula for coherent accumulation is as follows [49]

$$I(x,y) = \int_m s_{out}(t,m) \cdot \exp\left(j2\pi f_0 \tau_{i,j}(m)\right) dm \tag{19}$$

The energy of each pixel point in the grid is coherently superimposed over the target motion time. The pixel value is accumulated by each pixel point during the movement time to synthesize the final image.

3.2. GPU Acceleration of BP Algorithm for ISAR Imaging

Released by NVIDIA in 2006, CUDA is a general-purpose parallel computing platform and programming model built on GPUs. Computations for complex tasks can be performed more efficiently with CUDA programming. In recent years, CUDA programming techniques have been developed in hardware as well as software [50]. At the time of its initial release, CUDA was capable of utilizing GPUs with a limited number of cores, typically in the range of a few dozen or a few hundred. Consequently, it was not possible to make meaningful comparisons in terms of computational power between these early GPUs and the GPUs that are available today [51]. For example, NVIDIA's NVIDIA RTX 2080, which was released on 20 September 2018, is a GPU with 2944 CUDA cores and an FP32 compute power of 10.07 trillion times per second. However, the NVIDIA RTX 4090 now has 16,384 CUDA cores, with FP32 computing power reaching 82.58 trillion times per second. In recent years, CUDA has widely utilized its powerful parallel computing capabilities in the field of scientific computation [52].

In this paper, we realized the highly parallel BP imaging algorithm by CUDA. The scattered field data obtained by the SBR calculation are first loaded into the GPU Global Memory from the host memory under the <*cudaMemcpy*> function. These data contain information such as azimuth, angle of incidence, frequency, frequency sampling points, angle sampling points, polarization mode, etc. The specified memory space is allocated for these parameters from the GPU with memory size $N_f \times N_{phi} \times N_{fft}$ by the <*cudaMalloc*> function. Global variables on the device are defined through the *_device_* symbol, including the distance compression signal and the variables used to store the raw scattered field data, the variables used to store the coordinates and radar position, and the variables used to store the final imaging results. Range pulse compression of the raw echo data is performed using the <*cufft*> function library in CUDA, by which highly parallel FFTs and IFFTs can be realized. The range compression signal is copied into GPU memory using the <*cudaMemcpy*> function with an allocated memory size $N_f \times N_{phi} \times N_{fft}$.

Parallel accelerated computation is mainly implemented on the device by the CUDA kernel function. Kernel is an important concept in CUDA, and it is a function that is executed in parallel in a thread on the device. The kernel function is declared with the <_global_> symbol, and the number of threads required when calling this function needs to be specified, specifically by <<<grid, block>>>. The kernel function is executed by every thread. The azimuthal focusing process of the BP algorithm is written as a kernel function, and the corresponding number of threads are allocated to the kernel function to enable parallel computation on the GPU.

In Figure 10, the kernel function is responsible for calculating the distances of all pixels from the radar at each azimuthal moment, as well as phase compensation. The position of the corresponding azimuthal signal is determined by the distance of the pixel point from the radar. ISAR imaging using the BP algorithm is obtained by coherently superimposing the echoes of all pixel points at each azimuthal moment. The blocks and threads are defined as one-dimensional when running the CUDA function of the BP algorithm. The distribution of threads in each block is $(N_y, 1, 1)$, with N_x blocks in total and $(N_x, 1, 1)$ distribution of blocks. N_x and N_y are the number of pixels in the x direction and the number of pixels in the y direction, respectively. In order to optimize the efficiency of the kernel function in processing large-scale data, computational tasks for imaging pixels larger than 500×500 are batch-computed. The maximum number of pixel points computed in each batch is 500×500 in order to adapt to the resource limitations of the GPU. In the kernel function, the GPU allocates 250,000 threads for each batch of computational tasks, with every 500 threads being a block, for a total of 500 blocks. Each thread is responsible for computing the value of one-pixel point, and multiple batches of data will be executed in parallel on the CUDA core.

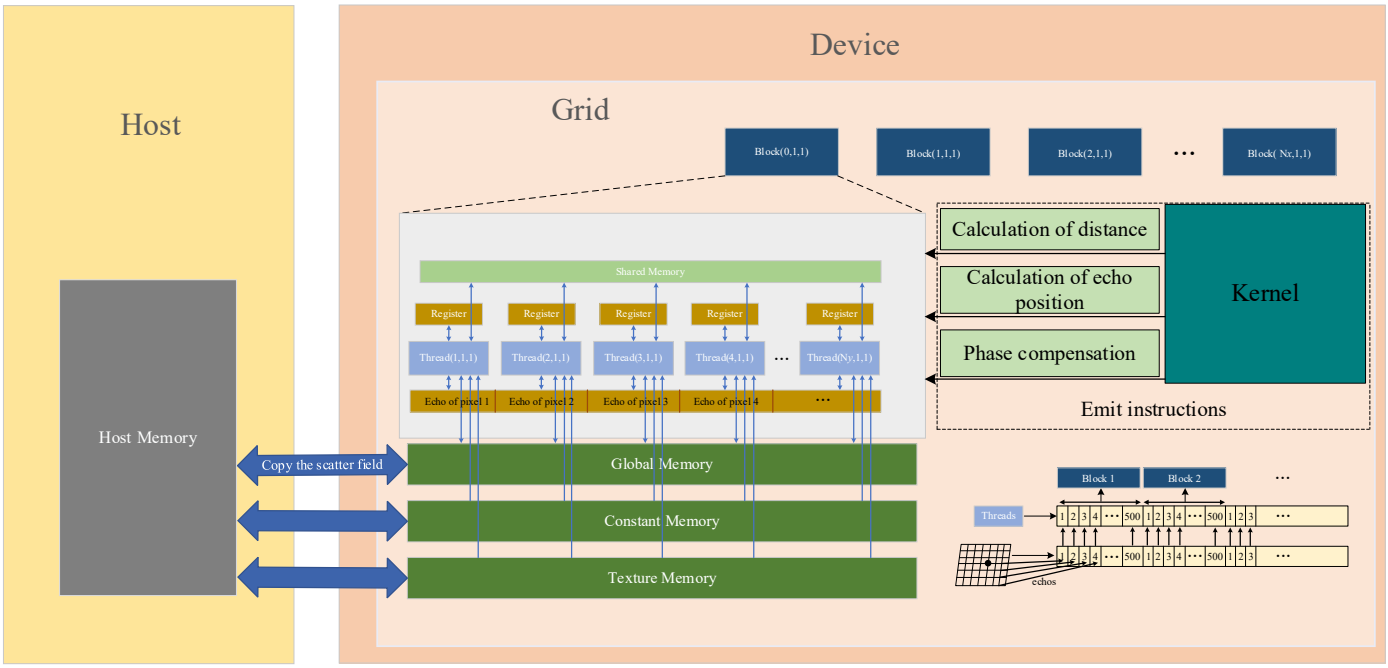


Figure 10. Parallel calculation process of the BP algorithm for ISAR imaging using CUDA acceleration.

In Figure 10, the scattered field data are first copied from Host Memory to the GPU Global Memory. Each thread can access the data stored in Global Memory. Constant Memory is a read-only memory used to broadcast instructions sent by the host. Texture Memory is used to store texture data. The number of blocks and threads is determined by the number of pixels N_x in the x direction and the number of pixels N_y in the y direction, respectively. In a block, all threads execute the same instructions, and each thread executes the kernel function. The kernel function relies on threads to realize the highly parallel computation of all the azimuthal focusing calculations.

Figure 11 presents the flow chart of the GPU-accelerated BP algorithm. After range compression, the imaging region is divided into a grid of pixels of size $N_f \times N_{phi}$. If the number of pixel grids in the imaging area is less than 500×500, then a corresponding number of threads are allocated to be responsible for computing the echo data of these pixel grids and accumulating them to the corresponding positions. The combination of these pixel grids is divided into several sub-regions for parallel processing if the pixels are over 500×500. Each thread of the GPU is responsible for processing a one-pixel point, including distance computation, phase compensation, etc. It calculates the current pixel location and accumulates the echo to the corresponding pixel grid. Once each block has completed the computation of its sub-region data, the data are transferred to their respective position in the final image. Finally, the entire image is output.

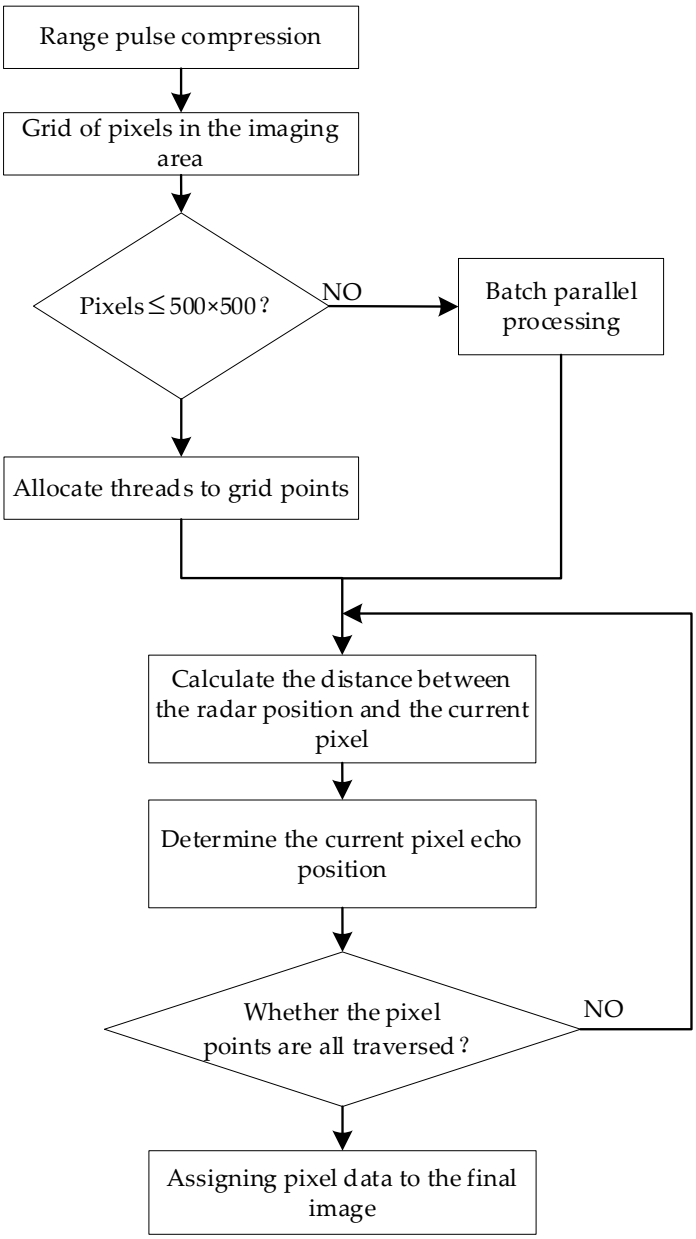


Figure 11. Flow chart of GPU-accelerated BP algorithm.

4. Results and Discussion

4.1. Validation of GPU-Accelerated SBR Method

The implementation of the GPU-accelerated electromagnetic computational method and the GPU-accelerated BP imaging method has been previously presented, as described in this paper. In this section, the validity of GPU-accelerated SBR combining GO with PO approximations is verified by comparison with RL-GO in FEKO v2020 software. Taking a full-scale F-22 fighter as an example, RL-GO is set up with two types of ray densities, one is $\lambda/10$ and the other is $\lambda/100$. A comparison of far-field RCS is made with the electromagnetic wave frequency $f_0 = 3$GHz. The model dissection produced 2444 triangular meshes, and the BVH tree was constructed to produce 4887 leaf nodes. The CAD model is as in Figure 12. The calculation time and memory cost are listed in Table 1. Our computer's CPU is a 12th Gen Intel(R) Core (TM) i5-12490F with a benchmark speed of 3.00 GHz, manufactured by Intel Corporation for China special edition products. The GPU is an NVIDIA GeForce RTX 2080 with 8.0 GB of dedicated GPU memory, manufactured by NVIDIA Corporation in China mainland.

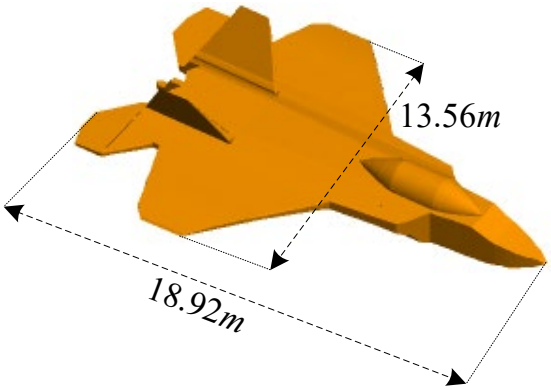


Figure 12. CAD model of full-scale F-22 fighter.

Table 1. A comparison of calculation time and memory cost for a full-scale F-22 fighter.

	Polarization	Calculation Time (s)			Memory (MB)		
		This Paper λ/10	RL-GO λ/10	RL-GO λ/100	This Paper λ/10	RL-GO λ/10	RL-GO λ/100
Figure 13a	VV	207.96	186.4	2303.3	557.6	130.9	181.4
Figure 13b	HH	433.195	103.1	1427.43	552.1	136.4	185.1

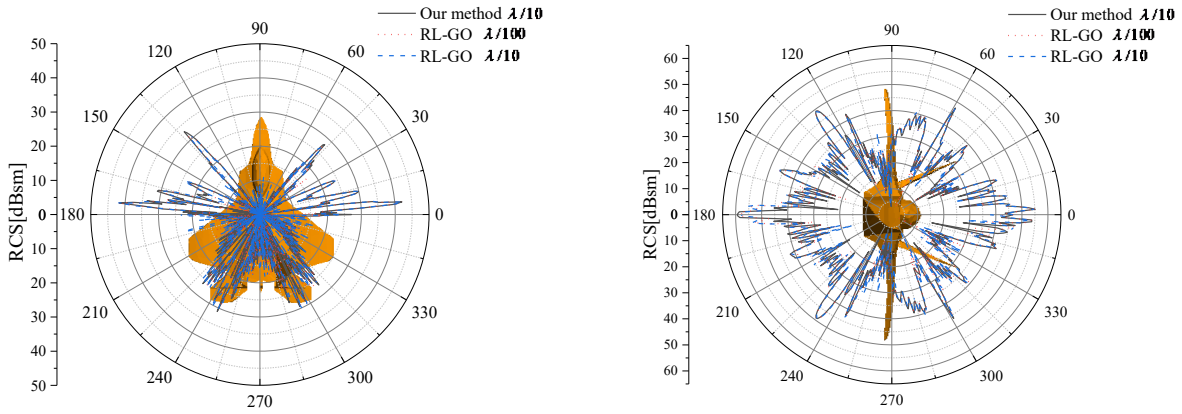


Figure 13. RCS angular distribution of a full-scale F-22 fighter: (**a**) incidence angle θ =90°, azimuth angle φ =0°∼ 360°, VV polarization; (**b**) incidence angle θ =0°∼ 360°, azimuth angle φ =0°, HH polarization.

Figure 13 shows the comparison of the RCS angular distribution of a full-scale F-22 fighter obtained by our GPU-accelerated SBR method and the parallel RL-GO method in FEKO. From Figure 13a,b, it is observed that the RCS calculated by our GPU-accelerated SBR method is generally in good agreement with that by the RL-GO method in FEKO. There is a slight difference between our GPU-accelerated SBR method and the parallel RL-GO method with a ray density of $\lambda/100$ in FEKO due to the effect of ray density, and the RL-GO method with a ray density of $\lambda/100$ also takes the diffraction field [53] into account.

4.2. ISAR Imaging Simulations

In this section, ISAR imaging results of representative aircraft target are presented and discussed. The matrix of the backscattered field of representative aircraft targets is calculated by our GPU-accelerated SBR method, and then the echoes are focused to obtain ISAR images by utilizing the GPU-accelerated BP algorithm developed in this paper. For

comparison, our GPU-accelerated BP algorithm was also applied to focus the backscattered field by employing the RL-GO method in the FEKO v2020 software to obtain focused ISAR.

Figure 14 shows the CAD model of a scaled A380 aircraft model with dimensions. This model has 3770 triangles. Figure 15a–c illustrate three typical observation configurations with different azimuthal scanning ranges. The incidence angle is fixed at $\theta = 60°$. The ISAR imaging parameters for Figures 16 and 18 are set as in Table 2.

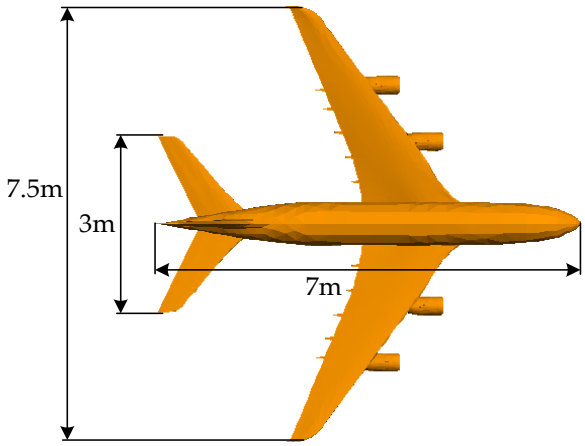


Figure 14. Computer-aided design (CAD) model and dimensions of a scaled A380 aircraft model.

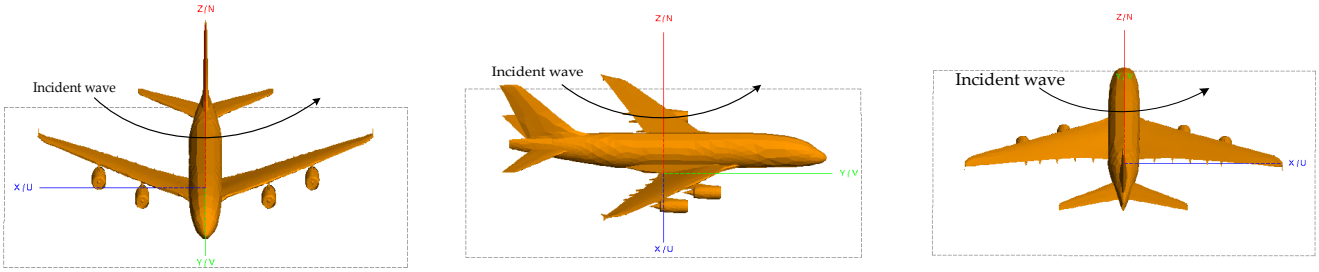


Figure 15. Three typical observation configurations with different azimuth angles under fixed incidence angle θ =60°. (**a**) $\varphi = 45° \sim 135°$; (**b**) $\varphi = -45° \sim 45°$; (**c**) $\varphi = -135° \sim -45°$.

Table 2. Parameters of ISAR simulation for Figures 16 and 18.

Parameter	Value
f_0	1.75 GHz
B	3 GHz
$\Delta\varphi$	90
θ	60 or 120
Δx	0.05 m
Δy	0.05 m
Sampling points	200
Polarization	VV

Figure 16 presents ISAR imaging results for three typical observation configurations with different azimuthal scanning ranges. The incidence angle is fixed at $\theta = 60°$. Our GPU-accelerated BP imaging algorithm is applied to focus the backscattering fields to obtain focused ISAR images. In Figure 16, Figure 16d–f are RL-GO at a ray density of $\lambda/10$ without diffraction and Figure 16g–i are RL-GO at a ray density of $\lambda/100$ with diffraction. Comparing Figures 16a–c and 16d–f, it can be found that the results of this paper's method and the RL-GO method are in better agreement with those obtained by FEKO's RL-GO

method under the condition of the same ray density, and it even outperforms the RL-GO at specific angles. For example, in Figure 16a,d, the difference in the scattering in the engine part of the airplane can be seen to be more obvious, and the differences in the scattering in the engine area of the airplane can be seen to be more obvious in Figure 16c,f; Figure 16f has strong clutter that overwhelms the information such as the structural features of the airplane, and the results of Figure 16f are not as good as the results of Figure 16c. When the ray density is $\lambda/100$, the echoes are able to record the geometrical structural features of the airplane in detail, so Figure 16g–i have very good imaging results compared to Figures 16a–c and 16d–f, which confirms the effectiveness of the GPU-accelerated BP imaging algorithms in this paper from the side. Table 3 shows the computation time and peak memory comparisons for Figure 16.

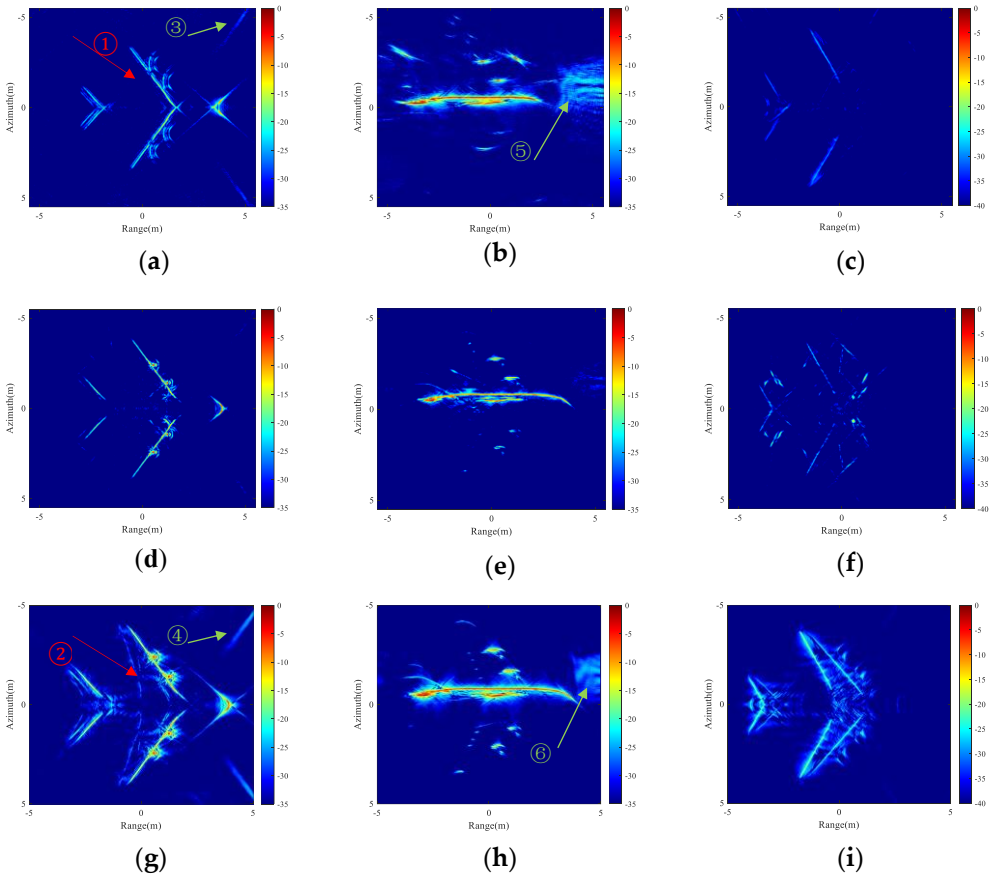


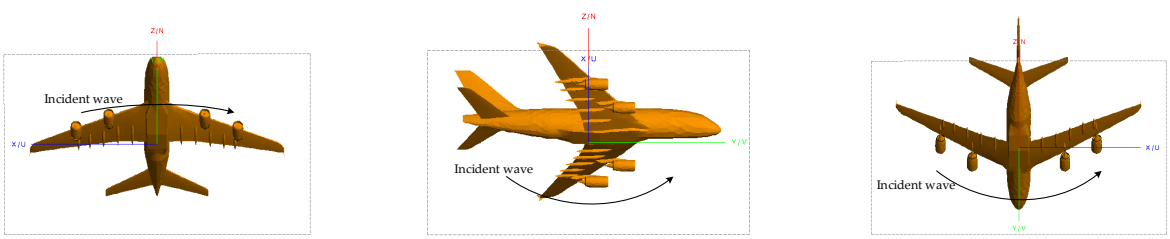
Figure 16. ISAR imaging results using GPU-accelerated BP imaging algorithm under fixed incidence angle $\theta = 60°$. In (**a**–**c**), the backscattering fields are calculated by our GPU-accelerated SBR method with a ray density of $\lambda/10$; in (**d**–**f**), the backscattering fields are obtained by FEKO's RL-GO method with a ray density of $\lambda/10$; in (**g**–**i**), the backscattering fields are obtained by RL-GO with a ray density of $\lambda/100$. (**a**,**d**,**g**) are results for azimuth angle φ =45° $\sim$ 135°; (**b**,**e**,**h**) are results for azimuth angle $\varphi = -45° \sim 45°$; (**c**,**f**,**i**) are results for azimuth angle $\varphi = -135° \sim -45°$.

Table 3. Computation time and peak memory for Figure 16.

	SBR		RL-GO	
	Time (s)	**Memory (MB)**	**Time (s)**	**Memory (MB)**
Figure 16a,d	2423.3	76.2	1584.2	177.8
Figure 16b,e	2669.8	78.4	1614.3	179.6
Figure 16c,f	2623.3	72.5	1474.1	179.5

It can be concluded that when the ray density is the same, the results of the method in this paper are in good agreement with those of RL-GO. Next, we analyze the difference between the results of the two methods when the ray densities are not the same. In Figure 16a, there are strong echoes from both the engine and the wing portion attached to the engine. The area indicated by the red arrow ① in (a) represents the wing position. At the observation angles φ =45° $\sim$ 135°, the ray will be reflected once after striking the wing portion at position ①. Due to the flatness of this portion of the structure, the ray will subsequently leave the target after being reflected. In our GPU-accelerated SBR method, multiple scattering is taken into account by ray tracing, in which the maximum reflection number is 10, while the diffraction field due to target edges is not taken into account, which leads to a pronounced difference between position ① in (a) and position ② in (g). Scattering spots appear at the positions indicated by the green arrows ③, ④, ⑤, and ⑥ in Figure 16a,b,g,h. Figure 16a,b are the results calculated by the method in this paper, and Figure 16g,h are the results calculated by the RL-GO with diffraction at a ray density of $\lambda/100$ in FEKO. Both our GPU-accelerated SBR and RL-GO in FEKO are ray-based methods. These scattering spots are related to the scattering mechanism of the electromagnetic waves and the ray tracing mechanism. Due to neglection of the diffraction field resulting from target edges, the target echo in Figure 16c is weaker than that in Figure 16i. A comparison of the ISAR imaging results of Figure 16a–c with Figures 16d–f and 16g–i demonstrates the feasibility and efficiency of our scheme by combining GPU-accelerated SBR with the BP algorithm for fast ISAR imaging simulation.

Figure 17a–c illustrate three typical observation configurations with different azimuthal scanning ranges. ISAR imaging results for three typical observation configurations with different azimuthal scanning ranges are shown in Figure 18a, b, and c, respectively. In Figure 18, the incidence angle is set as θ = 120°, and the other parameters are the same as those in Figure 16. In Figure 18a–c, the backscattering fields are calculated by our GPU-accelerated SBR method. In Figure 18d–i, the backscattering fields are obtained by the RL-GO method in FEKO v2020 software, where (d–f) are RL-GO at a ray density of $\lambda/10$ and (g–i) are RL-GO at a ray density of $\lambda/100$. Since the ray densities are the same and none of them include the wrap-around field, the results for the corresponding angle ranges in (a–c) and (d–f) are in good agreement. A comparison of the results of (c) and (f) shows that the echo strength of the airplane body is very weak in (f). There is also noise. Only the engine area exhibits strong echoes, which makes this paper's method work better under the same ray densities and it can better record the target's geometric structure information. In general, the focused ISAR images of backscattered fields calculated by our GPU-accelerated SBR method are in good agreement with those obtained by FEKO's RL-GO method. Some scattering spots can be observed as indicated by the red arrows in Figure 18a,b,g,h, as in Figure 16. This is due to multiple scattering of electromagnetic waves from the target surface. For rays reflected multiple times, the phase history of the electromagnetic wave is distorted. Therefore, the scattering spots indicated by the red arrows in Figure 18a,b,g,h can be eliminated by reducing the reflection numbers in the ray tracing algorithm. Table 4 shows the computation time and peak memory comparisons for Figure 18.



(a) (b) (c)

Figure 17. Three typical observation configurations with different azimuth angles under fixed incidence angle θ =120°. (**a**) $\varphi = 45° \sim 135°$; (**b**) $\varphi = -45° \sim 45°$; (**c**) $\varphi = -135° \sim -45°$.

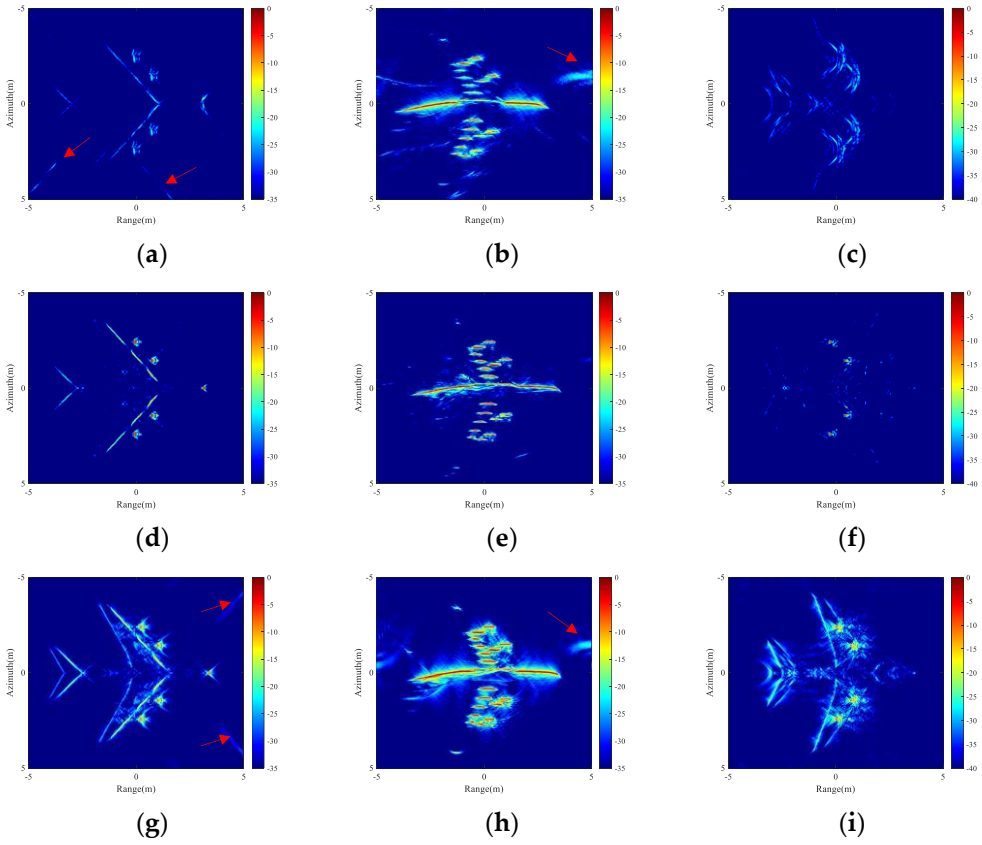


Figure 18. Similar to Figure 16 but with incidence angle $\theta = 120°$.

Table 4. Computation time and peak memory for Figure 18.

	SBR		RL-GO	
	Time (s)	**Memory (MB)**	**Time (s)**	**Memory (MB)**
Figure 18a,d	2315.2	72.4	1655.6	194.6
Figure 18b,e	2647.05	74.2	1702.4	202
Figure 18c,f	2872.9	77.1	1764.2	196.9

Figure 19 shows a CAD model of an electrically large-sized aircraft target, which is in the electrically large-sized category with an electrical size of 171 × 104 wavelengths. This CAD model has 712 face elements. ISAR images for this model were computed using the method in this paper and the RL-GO method in FEKO. The ray densities for both the method in this paper and the RL-GO method are set to be $\lambda/10$. The azimuth centers of Figure 20a–c are 90°, 0°, −90°. Figure 20 demonstrates three different azimuth ranges. In the following simulations of Figure 21, the ISAR imaging parameters are set as in Table 5.

Figure 21 shows the ISAR imaging results of both methods for electrically large-sized aircraft targets, with ray densities of $\lambda/10$ for this paper method and RL-GO in FEKO. Comparing Figures 21b and 21e, ① and ② in Figure 21b are the structural information of the airplane, and ① is the wing and ② is the tail, while Figure 21e cannot show this structural information, so this paper's method is better than the RL-GO imaging result when $\varphi = -45° \sim 45°$. Comparing Figures 21a and 21d, the result of Figure 21a is obviously better than that of Figure 21d, where phase history distortion occurs at the place indicated by the white arrows in Figure 21d. Detailed information on the wings, nose, and tail of the aircraft can be clearly displayed in Figure 21a. This also occurs in Figure 21c,f, where the white arrows in Figure 21c,f point. It can be clearly seen that the scattered spots that appear in these places indicated by the arrows are not aircraft structural information. It is a well-established fact that electromagnetic waves exhibit a multipath effect during

propagation. This phenomenon entails that an incident wave traverses a multitude of paths to reach a designated receiving point. Changes in phase history data caused by these different paths can be superimposed, resulting in phase history distortion. Table 6 shows the computation time and peak memory comparisons for Figure 21.

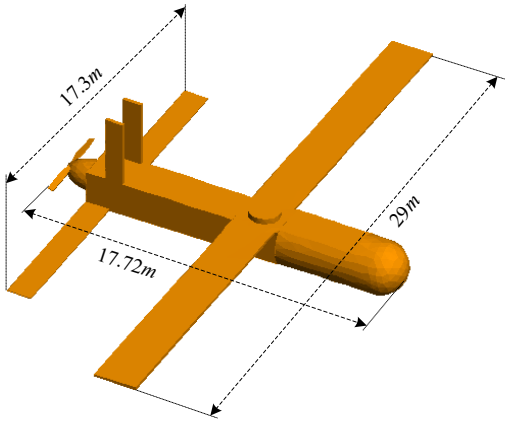
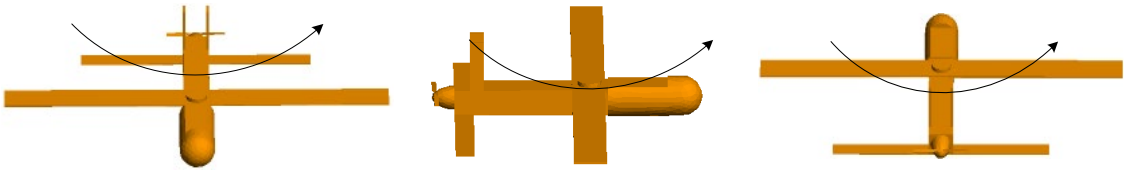


Figure 19. CAD model of an electrically large aircraft.



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

Figure 20. Three typical observation configurations with different azimuth angles under fixed incidence angle $\theta = 60°$. (**a**) $\varphi = 45° \sim 135°$; (**b**) $\varphi = -45° \sim 45°$; (**c**) $\varphi = -135° \sim -45°$.

Table 5. Parameters of ISAR simulation for Figure 21.

Parameter	Value
f_0	1.75 GHz
B	3 GHz
$\Delta\varphi$	90
θ	60
Δx	0.05 m
Δy	0.05 m
Sampling points	600
Polarization	VV

Table 6. Computation time and peak memory for Figure 21.

	SBR		RL-GO	
	Time (h)	**Memory (MB)**	**Time (h)**	**Memory (MB)**
Figure 21a,d	20.1	141.2	16.3	269.6
Figure 21b,e	19.6	140.7	15.5	258.5
Figure 21c,f	20.7	145.6	16.9	273.1

From numerical simulations of ISAR imaging, it is clearly indicated that strong scattering centers as well as target profiles can be observed under large observation azimuth angles and wide bandwidth. It is also indicated that the ISAR images are heavily sensitive to observation angles. Due to multiple scattering, several triangular patches will be

hit by identical rays, resulting in the phase history distortion of electromagnetic waves. Phase history distortion is a common problem with ray methods. Thus, obvious sidelobes can be observed in focused ISAR images. In comparison with RL-GO in FEKO v2020 software, the feasibility and efficiency of our scheme are demonstrated by combining GPU-accelerated SBR with BP algorithm for fast ISAR imaging simulations under wide-angle and wide-bandwidth conditions.

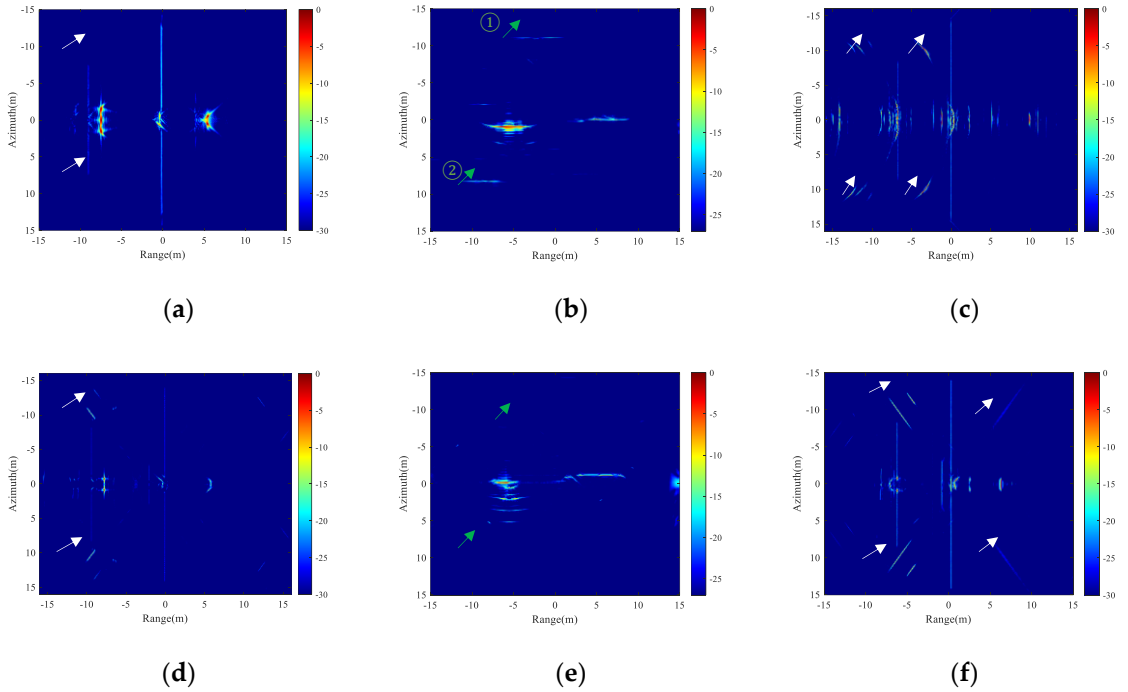


Figure 21. ISAR imaging results using GPU-accelerated BP imaging algorithm under fixed incidence angle $\theta = 60^\circ$ in (**a**–**c**), the backscattering fields are calculated by our GPU-accelerated SBR method; in (**d**–**f**), backscattering fields are obtained by FEKO's RL-GO method at a ray density of $\lambda/10$; (**a**,**d**) are results for azimuth $\varphi = 45^\circ \sim 135^\circ$; (**b**,**e**) are results for azimuth $\varphi = -45^\circ \sim 45^\circ$; (**c**,**f**) are results for azimuth $\varphi = -135^\circ \sim -45^\circ$, VV polarization.

5. Conclusions

In this paper, a novel bouncing ray method based on GPU and BVH tree acceleration is presented for ISAR imaging simulations. It employs C++AMP to achieve GPU parallel acceleration computation, in which a BVH tree structure is constructed according to the target structure, thereby enhancing ray intersection efficiency. The SAH method is incorporated into the scene bounding box division, effectively mitigating the impact of bounding box overlapping on the ray traversal efficiency of the BVH tree structure. To efficiently perform ISAR imaging simulations, a GPU-based accelerated BP imaging algorithm has been developed by virtue of CUDA. The accuracy of GPU-accelerated SBR is validated by comparing the RCS calculated by our SBR method with that obtained by RL-GO in FEKO. It is demonstrated that the presented GPU-accelerated SBR shows good validity and reliability. For ISAR imaging simulations, taking an A380 and a simplified aircraft model as examples, the backscattering fields were calculated utilizing the GPU-accelerated SBR algorithm under large azimuth angles, $\Delta\varphi = 90^\circ$, and wide bandwidths, 3 GHz. The backscattered echoes are focused using the GPU-accelerated BP imaging algorithm, and the focused ISAR images of our GPU-accelerated SBR method are in good agreement with those of FEKO's RL-GO method, indicating the feasibility and efficiency of our GPU-accelerated BP ISAR imaging algorithm. The simulations indicate that strong scattering centers as well as target profiles can be observed clearly from ISAR images under large observation angles and wide bandwidths. Obvious sidelobes in focused ISAR images can be observed, due

to the phase history of electromagnetic waves being distorted resulting from multipole scattering. It is also indicated by numerical simulations that the ISAR imaging results are heavily sensitive to observation angles. In the future, the present work will be extended to swarm targets' ISAR imaging as well as 3D ISAR imaging by developing an efficient electromagnetic simulation algorithm by combining GPU-accelerated SBR and PTD for taking edge diffraction into account.

Author Contributions: Methodology, J.C., R.Z., P.Y., and R.W.; software, J.C.; validation, J.C.; formal analysis, J.C.; ray tracing simulation, J.C.; investigation, J.C. and P.Y.; resources, P.Y.; writing—original draft preparation, J.C.; writing—review and editing, P.Y.; visualization, J.C.; supervision, P.Y.; project administration, J.C. and P.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China [Grant Nos. 62061048, 62261054, and 62361054], in part by the Shaanxi Key Research and Development Program [Grant Nos. 2023-YBGY-254, and 2024GXYBXM-108], in part by the Natural Science Basic Research Plan in Shaanxi Province of China (Grant No. 2023-JC-YB-539), and in part by the Graduate Education Innovation Program of Yan'an University (Grant No. YCX2024086).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors thank the editors and reviewers for their constructive suggestions.

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

Abbreviations

The following abbreviations are used in this manuscript:

BP	Back Projection
BVH	Bounding Volume Hierarchies
C++AMP	C++ Accelerated Massive Parallelism
CUDA	Compute Unified Device Architecture
CAD	Computer-Aided Design
CPU	Central Processing Unit
EDM	Equivalent Dipole Moment
FFT	Fast Fourier Transform
GPU	Graphics Processing Unit
GO	Geometrical Optics
ISAR	Inverse Synthetic Aperture Radar
IFFT	Inverse Fast Fourier Transform
Kd-tree	K-dimensional tree
MOM	Method of Moments
PTD	Physical Theory of Diffraction
PO	Physical Optics
PEC	Perfect Electrical Conductor
RCS	Radar Cross Section
RL-GO	Ray Launching Geometrical Optics
RMSE	Root Mean Squared Error
SBR	Shooting and Bouncing Ray
SAH	Surface Area Heuristic
SMs	Streaming Multiprocessors
SPs	Streaming Processors
SIMT	Single Instruction, Multiple Threads
TDSBR	Time-Domain Shooting and Bouncing Ray
TDPO	Time-Domain Physical Optics

References

1. Bhalla, R.; Ling, H. ISAR image formation using bistatic data computed from the shooting and bouncing ray technique. *J. Electromagn. Waves Appl.* **1993**, *7*, 1271–1287. [CrossRef]
2. He, X.Y.; Zhou, X.Y.; Cui, T.J. Fast 3D-ISAR image simulation of targets at arbitrary aspect angles through nonuniform fast Fourier transform (NUFFT). *IEEE Trans. Antennas Propag.* **2012**, *60*, 2597–2602. [CrossRef]
3. Zhang, K.; Wang, C.F.; Jin, J.M. Broadband Monostatic RCS and ISAR Computation of Large and Deep Open Cavities. *IEEE Trans. Antennas Propag.* **2018**, *66*, 4180–4193. [CrossRef]
4. Prickett, M.; Chen, C. Principles of inverse synthetic aperture radar/ISAR/imaging. In Proceedings of the EASCON 1980, Electronics and Aerospace Systems Conference, Arlington, VA, USA, 29 September–1 October 1980; pp. 340–345.
5. García-Fernández, A.F.; Yeste-Ojeda, O.A.; Grajal, J. Facet Model of Moving Targets for ISAR Imaging and Radar Back-Scattering Simulation. *IEEE Trans. Aerosp. Electron. Syst.* **2010**, *46*, 1455–1467. [CrossRef]
6. Lee, J.I.; Yun, D.J.; Kim, H.J.; Yang, W.Y.; Myung, N.H. Fast ISAR Image Formations Over Multiaspect Angles Using the Shooting and Bouncing Rays. *IEEE Antennas Wirel. Propag. Lett.* **2018**, *17*, 1020–1023. [CrossRef]
7. Guo, G.; Guo, L.; Wang, R.; Li, L. An ISAR Imaging Framework for Large and Complex Targets Using TDSBR. *IEEE Antennas Wirel. Propag. Lett.* **2021**, *20*, 1928–1932. [CrossRef]
8. Meng, W.; Li, J.; Xi, Y.J.; Guo, L.X.; Li, Z.H.; Wen, S.K. An Improved Shooting and Bouncing Ray Method Based on Blend-Tree for EM Scattering of Multiple Moving Targets and Echo Analysis. *IEEE Trans. Antennas Propag.* **2024**, *72*, 2723–2737. [CrossRef]
9. Yang, P.J.; Wu, R.; Ren, X.C.; Zhang, Y.Q.; Zhao, Y. Doppler spectra of electromagnetic wave scattered from an object flying above time-varying nonlinear sea surfaces. *J. Electromagn. Waves Appl.* **2019**, *33*, 2175–2198. [CrossRef]
10. Ling, H.; Chou, R.C.; Lee, S.W. Shooting and bouncing rays: Calculating the RCS of an arbitrarily shaped cavity. *IEEE Trans. Antennas Propag.* **1989**, *37*, 194–205. [CrossRef]
11. Xu, F.; Jin, Y.Q. Bidirectional analytic ray tracing for fast computation of composite scattering from electric-large target over a randomly rough surface. *IEEE Trans. Antennas Propag.* **2009**, *57*, 1495–1505. [CrossRef]
12. Tao, Y.; Lin, H.; Bao, H. GPU-Based Shooting and Bouncing Ray Method for Fast RCS Prediction. *IEEE Trans. Antennas Propag.* **2010**, *58*, 494–502. [CrossRef]
13. Meng, W.; Li, J.; Guo, L.X.; Yang, Q.J. An Accelerated SBR Method for RCS Prediction of Electrically Large Target. *IEEE Antennas Wirel. Propag. Lett.* **2022**, *21*, 1930–1934. [CrossRef]
14. Baden, J.M.; Tripp, V.K. Ray reversal in SBR RCS calculations. In Proceedings of the 2015 31st International Review of Progress in Applied Computational Electromagnetics (ACES), Williamsburg, VA, USA, 22–26 March 2015; pp. 1–2.
15. Feng, T.T.; Guo, L.X. An Improved Ray-Tracing Algorithm for SBR-Based EM Scattering Computation of Electrically Large Targets. *IEEE Antennas Wirel. Propag. Lett.* **2021**, *20*, 818–822. [CrossRef]
16. Yun, K.C.; Fu, W.C. Efficient GPU Implementation of the High-Frequency SBR-PO Method. *IEEE Antennas Wirel. Propag. Lett.* **2013**, *12*, 941–944. [CrossRef]
17. Wu, R.; Wu, B.Y.; He, P.X.; Guo, K.Y.; Sheng, X.Q. A Fast Plane Wave Expansion Algorithm for Rigorous Scattering Analysis from Swarm Targets. *IEEE Trans. Antennas Propag.* **2023**, *71*, 7426–7437. [CrossRef]
18. Dong, C.-L.; Guo, L.-X.; Meng, X. An accelerated algorithm based on GO-PO/PTD and CWMFSM for EM scattering from the ship over a sea surface and SAR image formation. *IEEE Trans. Antennas Propag.* **2020**, *68*, 3934–3944. [CrossRef]
19. Dong, C.-L.; Guo, L.-X.; Meng, X.; Wang, Y. An accelerated SBR for EM scattering from the electrically large complex objects. *IEEE Antennas Wirel. Propag. Lett.* **2018**, *17*, 2294–2298. [CrossRef]
20. Meng, W.; Li, J.; Chai, S.R.; Xi, Y.J.; Wen, S.K.; Liu, R.F. An Improved SBR-PTD Method for EM Scattering from Moving Target. In Proceedings of the 2023 International Applied Computational Electromagnetics Society Symposium (ACES-China), Hangzhou, China, 15–18 August 2023; pp. 1–3.
21. Li, H.Z.; Dong, C.L.; Meng, X.; Guo, L.X.; Wei, Q.H. A Novel Equivalent Dipole Moment-Based MoM-SBR Hybrid Method for EM Scattering Computation of Electrically Large Complex Targets. In Proceedings of the 2023 International Conference on Microwave and Millimeter Wave Technology (ICMMT), Qingdao, China, 14–17 May 2023; pp. 1–3.
22. Wang, Z.; Wei, F.; Huang, Y.; Zhang, X.; Zhang, Z. Improved RD imaging method based on the principle of step-by-step calculation. In Proceedings of the 2021 2nd China International SAR Symposium (CISS), Shanghai, China, 3–5 November 2021; pp. 1–5.
23. Dong, L.; Han, S.; Zhu, D.; Mao, X. A Modified Polar Format Algorithm for Highly Squinted Missile-Borne SAR. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 1–5. [CrossRef]
24. Boag, A.G.A. Backprojection Imaging of Moving Objects. *IEEE Trans. Antennas Propag.* **2021**, *69*, 4944–4954. [CrossRef]
25. Zhang, M.; Ren, Z.; Zhang, G.; Zhang, C. THz ISAR imaging using GPU-accelerated phase compensated back projection algorithm. *J. Infrared Millim. Waves* **2022**, *41*, 448–456. [CrossRef]
26. Arikan, O.; Munson, D.C., Jr. A New Back-Projection Algorithm for Spotlight-Mode SAR and ISAR. In Proceedings of the High Speed Computing II, Los Angeles, CA, USA, 17–18 January 1989; pp. 107–117.
27. Gong, H.; Liu, Y.; Chen, X.; Wang, C. Scene optimization of GPU-based back-projection algorithm. *J. Supercomput.* **2023**, *79*, 4192–4214. [CrossRef]
28. Guo, G.; Guo, L.; Wang, R.; Liu, W.; Li, L. Transient Scattering Echo Simulation and ISAR Imaging for a Composite Target-Ocean Scene Based on the TDSBR Method. *Remote Sens.* **2022**, *14*, 1183. [CrossRef]

29. Sun, T.-P.; Cong, Z.; He, Z.; Ding, D. An Accelerated Time-Domain Iterative Physical Optics Method for Analyzing Electrically Large and Complex Targets. *Electronics* **2022**, *12*, 59. [CrossRef]
30. Guo, G.; Guo, L.; Wang, R. ISAR Image Algorithm Using Time-Domain Scattering Echo Simulated by TDPO Method. *IEEE Antennas Wirel. Propag. Lett.* **2020**, *19*, 1331–1335. [CrossRef]
31. Zhou, J.; Han, Y. Analyzing the electromagnetic scattering characteristics for plasma targets based on shooting and bouncing ray method. *AIP Adv.* **2019**, *9*, 065106. [CrossRef]
32. Gordon, W.B. High frequency approximations to the physical optics scattering integral. *IEEE Trans. Antennas Propag.* **1994**, *42*, 427–432. [CrossRef]
33. Fan, T.T.; Zhou, X.; Yu, W.M.; Zhou, X.Y.; Cui, T.J. Time-Domain Line-Integral Representations of Physical-Optics Scattered Fields. *IEEE Trans. Antennas Propag.* **2017**, *65*, 309–318. [CrossRef]
34. Liao, C. Research on Nearfield Scattering Modeling of Ship on the Sea Surface Based on high Frequency Method (In Chinese). Master's Thesis, University of Electronic Science and Technology of China, Chengdu, China, 2021. Available online: https://kns.cnki.net/kcms2/article/abstract?v=n6BwBobH4uvU7PG733EVJdhS9-f9LApXUEAHzK60Kgv6ciotFWwf11njOZZqPyjAOLTnfvU-beMgAqMR8blHbC9mFOJ0F5tkD-vf1xHSqT6eY_XonBN7ouPAQRKMghtFziV-7qPBjXZhfcEiaH_takq8r-_JoHJuM61BQbTbKyScQeIPY8_hM3AAmLBaJTfo9XIwNvOUOSI=&uni (accessed on 29 July 2024).
35. Stratton, J.A.; Chu, L. Diffraction theory of electromagnetic waves. *Phys. Rev.* **1939**, *56*, 99. [CrossRef]
36. Chu, L.J.; Stratton, J.A. Elliptic and spheroidal wave functions. *J. Math. Phys.* **1941**, *20*, 259–309. [CrossRef]
37. He, X.-Y.; Wang, X.-B.; Zhou, X.; Zhao, B.; Cui, T.-J. Fast ISAR image simulation of targets at arbitrary aspect angles using a novel SBR method. *Prog. Electromagn. Res. B* **2011**, *28*, 129–142. [CrossRef]
38. Guo, G.; Guo, L.; Wang, R. The Study on Near-Field Scattering of a Target Under Antenna Irradiation by TDSBR Method. *IEEE Access* **2019**, *7*, 113476–113487. [CrossRef]
39. Tang, X.; Feng, Y.; Gong, X. Mo M-PO/SBR Algorithm Based on Collaborative Platform and Mixed Model. *Trans. Nanjing Univ. Aeronaut. Astronaut.* **2019**, *36*, 589–598. [CrossRef]
40. Li, J.; Meng, W.; Chai, S.; Guo, L.; Xi, Y.; Wen, S.; Li, K. An Accelerated Hybrid Method for Electromagnetic Scattering of a Composite Target–Ground Model and Its Spotlight SAR Image. *Remote Sens.* **2022**, *14*, 6632. [CrossRef]
41. Zhu, R. Speedup of Micromagnetic Simulations with C++ AMP on Graphics Processing Units. *Comput. Sci. Eng.* **2016**, *18*, 53–59. [CrossRef]
42. Sihai, W.; Hu, Z.; Haotian, P.; Lu, C. Accelerated Parallelism in Numerical Simulation with C++ AMP. In Proceedings of the 2016 Workshop: Workshop High Performance Computing, Beijing, China, 14–16 November 2016; pp. 53–55.
43. Wynters, E. Fast and easy parallel processing on GPUs using C++ AMP. *J. Comput. Sci. Coll.* **2016**, *31*, 27–33.
44. Shyamala, K.; Kiran, K.R.; Rajeshwari, D. Design and implementation of GPU-based matrix chain multiplication using C++AMP. In Proceedings of the 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 22–24 February 2017; pp. 1–6.
45. Damkjær, J. *Stackless BVH Collision Detection for Physical Simulation*; University of Copenhagen Universitetsparken: København, Denmark, 2007; Available online: http://image.diku.dk/projects/media/jesper.damkjaer.07.pdf (accessed on 29 July 2024).
46. Chung, S.; Choi, M.; Youn, D.; Kim, S. Comparison of BVH and KD-tree for the GPGPU acceleration on real mobile devices. In Proceedings of the Frontier Computing: Theory, Technologies and Applications (FC 2018), Kyushu, Japan, 9–12 July 2019; pp. 535–540.
47. Sopin, D.; Bogolepov, D.; Ulyanov, D. Real-time SAH BVH construction for ray tracing dynamic scenes. In Proceedings of the Графикон'2011, Moscow, Russia, 26–30 September 2011; pp. 74–77.
48. Huipeng, Z.; Junling, W.; Di, X.; Xiaoyang, Q. The modified back projection algorithm for bistatic ISAR imaging of space objects. In Proceedings of the 2016 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Hong Kong, China, 5–8 August 2016; pp. 1–5.
49. Xiao, D. *Study on ISAR Imaging of Space Targets Using BP Technology*; Nanjing University: Nanjing, China, 2017.
50. Pu, L.; Zhang, X.; Yu, P.; Wei, S. A fast three-dimensional frequency-domain back projection imaging algorithm based on GPU. In Proceedings of the 2018 IEEE Radar Conference (RadarConf18), Oklahoma City, OK, USA, 23–27 April 2018; pp. 1173–1177.
51. Li, Z.; Qiu, X.; Yang, J.; Meng, D.; Huang, L.; Song, S. An Efficient BP Algorithm Based on TSU-ICSI Combined with GPU Parallel Computing. *Remote Sens.* **2023**, *15*, 5529. [CrossRef]
52. Afif, M.; Said, Y.; Atri, M. Computer vision algorithms acceleration using graphic processors NVIDIA CUDA. *Clust. Comput.* **2020**, *23*, 3335–3347. [CrossRef]
53. Ufimtsev, P.Y. Elementary edge waves and the physical theory of diffraction. *Electromagnetics* **1991**, *11*, 125–160. [CrossRef]

electronics

MDPI

Article

Unlocking Business Value: Integrating AI-Driven Decision-Making in Financial Reporting Systems

Alin Emanuel Artene [1,*], Aura Emanuela Domil [2] and Larisa Ivascu [1]

[1] Management Department, Faculty of Management in Production and Transportation, Politehnica University of Timisoara, 14 Remus Street, 300009 Timisoara, Romania; larisa.ivascu@upt.ro
[2] Department of Accounting and Audit, Faculty of Economics and Business Administration, West University of Timisoara, 300223 Timisoara, Romania; aura.domil@e-uvt.ro
* Correspondence: alin.artene@upt.ro

Abstract: This research article investigates the synergies between artificial intelligence (AI), digital transformation (DT), and financial reporting systems within the business context. The central theme explores how organizations enhance their decision-making processes by integrating AI technologies into digital transformation initiatives, particularly in financial reporting. The focal point is comprehending how the synergy of these integrated systems can unlock substantial business value, instigate strategic innovation, and elevate overall financial analytics through the adoption of intelligent, data-driven decision-making methodologies. By harnessing advanced analytics, automation, and adaptive decision support capabilities, organizations navigate the complexities of a rapidly evolving business environment, in which neural networks emerge as a valuable tool for calibrating outcomes in the financial accounting environment, demonstrating effectiveness in processing complex financial data, identifying patterns, and making predictions, ushering in a new era of transformative possibilities. The introduction of a game theory payoff matrix in this AI decision-making tool adds a strategic framework for analyzing interactions among decision-makers, considering strategic choices and outcomes in a dynamic and competitive context.

Keywords: digital transformation; decision-making systems; integrated systems; neural networks; business financial reporting; game theory payoff matrix

Citation: Artene, A.E.; Domil, A.E.; Ivascu, L. Unlocking Business Value: Integrating AI-Driven Decision-Making in Financial Reporting Systems. *Electronics* **2024**, *13*, 3069. https://doi.org/10.3390/electronics13153069

Academic Editor: Domenico Ursino

Received: 12 July 2024
Revised: 24 July 2024
Accepted: 30 July 2024
Published: 2 August 2024

1. Introduction

Automation, artificial intelligence, and data analysis lead to significant changes in the way financial institutions operate. In the ever-evolving landscape of financial reporting, organizations face an urgent imperative to harness the transformative potential of digital technologies. The impact of digitization on finance is undeniable. The paradigm shift towards digital transformation represents a critical point where companies must not only adapt but also innovate to remain competitive in the global marketplace. Central to the objectives of this evolution is the integration of artificial intelligence (AI) technologies, which offer unprecedented opportunities to revolutionize strategic management within economic financial reporting systems. The control process, in this paradigm, evolves from rigid surveillance to dynamic orchestration.

Embedded AI systems equipped with sophisticated algorithms demonstrate the ability to continuously learn from data streams, adapt to evolving patterns, and make real-time decisions. This presents a shift from a deterministic control model to a more adaptive and probabilistic framework where control is exercised through algorithmic governance and continuous feedback loops.

In the era of post-pandemic digital transformation, the integration of artificial intelligence technologies is a cornerstone for public and private economic entities that aspire to navigate the complex landscape of financial reporting and economic productivity with

precision and agility. The convergence of artificial intelligence and the digitization of financial reporting systems [1] does not only represent a technological paradigm shift but also introduces a profound redefinition of control mechanisms and management processes. Integrated AI systems are reshaping control frameworks and empowering decision-makers in the dynamic context of financial reporting amid digital transformation.

As businesses embark on their digital transformation and digitization journey [2], the strategic and subsidized infusion of AI technologies holds the promise of unlocking unprecedented business value. Traditional financial reporting systems, rooted in deterministic methodologies, are now at a crossroads as organizations seek to incorporate advanced artificial intelligence algorithms. The integration of machine learning, predictive analytics, and cognitive computing into these systems introduces a new dimension of autonomy and adaptability. As AI becomes an active participant in decision-making processes, the contours of control mechanisms must be reevaluated.

Traditional hierarchical models are challenged by the distributed nature of decision-making in embedded AI systems [3], where algorithms learn, adapt, and contribute autonomously. The need to understand the complexities and implications of this integration has become paramount, particularly in terms of its impact on financial reporting decision-making. Traditional reporting systems, despite being an integral part of organizational functioning, now face the imperative to evolve. The following question arises: how can judicious integration of AI technologies improve decision-making processes in the context of financial reporting during the digital transformation process?

Figure 1 shows AI-driven decision-making in digital transformation. This includes digital transformation initiatives, AI integration, decision-making enhancement, benefits, use cases, consideration, future trends, implementation strategies, and measurement metrics. Each of these directions is important in the evolution of digitization. Evaluating each direction, the following can be stated:

- Digital transformation initiatives: these represent steps that contribute to DT. Components that contribute to the success of the DT process, strategic elements, and opportunities and challenges may be included.
- AI integration: this includes defining how AI has an important role in DT, the different types of associated technologies, and the implementation of opportunities and solutions. All this contributes to important support in the DT process.
- Decision-making enhancement: this refers to the use of data to gain competitive advantages, predictive analytics, and automation of organizational processes, and streamline processes through effective decision-making.
- Benefits: the benefits identified by the beneficiaries are multiple and among them are the improvement of accuracy, time and cost efficiency, the improvement of the customer's experience, and other related benefits.
- Example: evaluating the specific elements in which AI can be used, we can mention financial reporting, operation management, customer-facing the applications of AI, and many others.
- Consideration: the important elements that must be evaluated in this approach are the ethical implications, security, and human–AI collaboration.
- Future trends: this field is growing considerably, and future approaches should be anticipated at the organizational level through well-thought-out strategies and approaches (emerging trends, evolution of the field, and technological approaches).
- Implementation strategies: the elements of strategic management are opportune at the organizational level for good alignment with technological advancement.
- Measurement metrics: this entire approach must be measured, and the performance and success indicator metrics for AI implementation include correct organizational approaches.

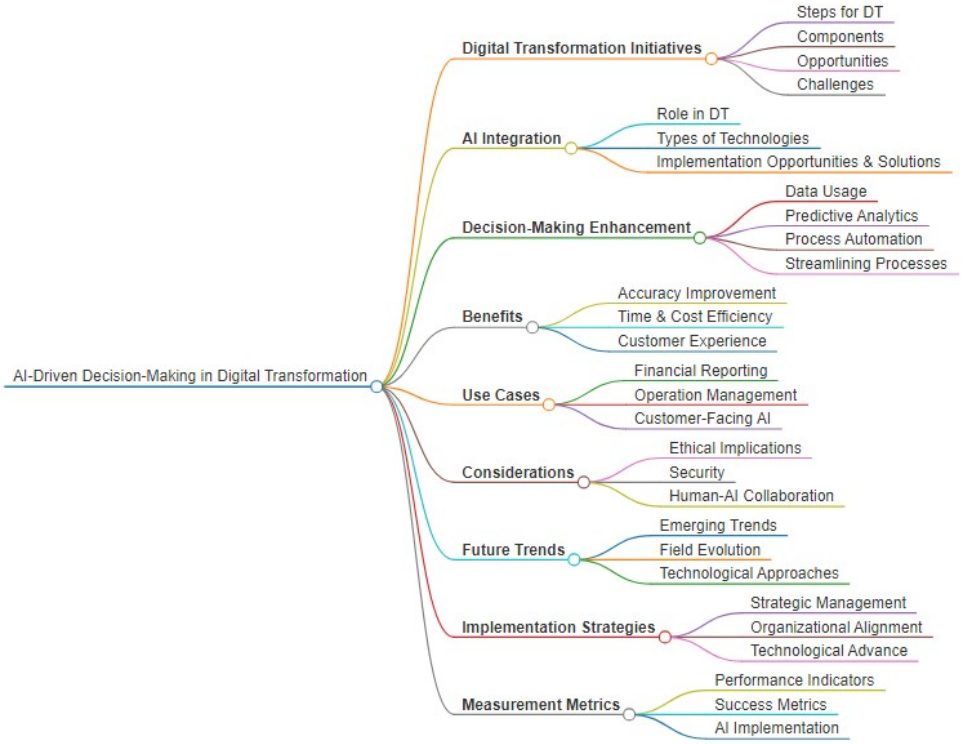


Figure 1. AI driven decision-making mind map.

In a landscape where data-driven insights drive organizational success, the potential benefits of AI—from streamlining processes to discovering actionable intelligence—invite organizations to explore new frontiers in decision-making capabilities. Against this background, this article undertakes an exploration of the transformative role that AI technologies are playing in reshaping the financial reporting decision-making landscape in the age of digital transformation. Delving into the complicated interplay between AI integration and the evolving nature of financial reporting systems, we seek to provide nuanced insights that connect theoretical advances with practical implications. Our goal is to guide organizations to a comprehensive understanding of how AI, when seamlessly integrated, can not only meet but exceed decision-making needs in the dynamic field of financial reporting in the digital age.

2. Literature Review

The integration of artificial intelligence (AI) into financial reporting systems has gained significant attention in the recent literature [4] due to its potential to transform traditional financial processes and unlock business value. The authors of [5] investigate the impact of integrating artificial intelligence into accounting information systems for enhancing non-financial performance in manufacturing companies, revealing significant positive effects on both the efficiency of accounting information systems and overall non-financial performance. Numerous studies have emphasized the role of AI in enhancing financial decision-making processes. Ref. [1] demonstrated the effectiveness of machine learning algorithms in predicting financial trends, optimizing resource allocation, and providing valuable insights for strategic planning. The integration of neural networks, as discussed by [6], has shown promise in automating tasks, improving forecasting accuracy, and supporting decision-makers in the financial domain.

The role of big data and leveraging AI-driven decision-making and digital transformation in financial reporting systems unlocks business value, aligning with stakeholder preferences, and addressing challenges and opportunities in the evolving financial landscape [5]. In the context of the modern economy, the transformative impact of big data extends beyond its common use by economists, prompting a shift in focus toward understanding

how others' utilization of data influences market outcomes [5]. This exploration involves incorporating big data into contemporary economic and financial theories. Notably, one application involves leveraging big data to enhance the decision-making of financial market participants, influencing firm prices, cost of capital, and investment dynamics.

The concept of digital transformation in financial reporting has been explored extensively in the literature. Scholars such as the authors of [7,8] discussed the evolution of financial reporting systems from manual processes to advanced digital platforms. The shift towards cloud-based solutions, the automation of data entry tasks, and the use of advanced analytics tools have been identified as key components of digital transformation in financial reporting [9].

Research by Sorensen [10] investigates the importance of aligning financial decisions with stakeholder preferences. Ref. [10] highlights the need for tools that consider the perspectives of various stakeholders, such as Company Management and Shareholders. The integration of AI-driven decision-making tools, informed by neural networks, offers a mechanism to align strategic choices with the preferences of key stakeholders.

The literature also recognizes challenges associated with the integration of AI and digital transformation in financial reporting. Concerns related to data privacy, regulatory compliance, and the need for skilled professionals capable of navigating the intersection of finance and AI have been discussed [3,11]. However, studies also highlight the opportunities for cost savings, efficiency improvements, and strategic advantages that arise from successful implementation [12]. Numerous studies emphasize the importance of financial KPIs in performance measurement and strategic decision-making within organizations [13] Metrics such as return on investment (ROI), earnings per share (EPS), and profit margin are recognized as critical indicators of financial health. Researchers argue that incorporating these KPIs into neural networks can enhance the accuracy of financial predictions and decision support systems [14,15].

Examining the intersection of game theory and financial decision-making [16] explores how strategic interactions among market participants, influenced by information asymmetry and competition, impact financial outcomes. The study may not directly focus on KPIs but provides insights into decision dynamics. While there might not be an extensive body of literature explicitly combining financial KPIs, decision-making, and game theory, some studies provide a foundation for understanding the interconnectedness of these concepts. Addressing the transparency and interpretability of AI decision models, Ref. [17] discusses the importance of explainability to gain the trust and acceptance of AI tools in decision-making, especially in sensitive domains like healthcare and finance.

The literature review indicates a growing body of research emphasizing the diverse applications, challenges, and ethical considerations associated with the integration of AI tools in decision-making. Understanding the impact of AI across various domains provides a foundation for future developments in creating a more effective, transparent, and ethical decision-support system.

In recent years, accounting and audit companies have tapped into the potential of AI to revolutionize traditional processes within financial institutions and have developed applications of machine learning in finance. JP Morgan Chase, a global leader in financial services, developed the Contract Intelligence, or COiN, platform, an AI-driven tool designed to review legal documents and extract critical data points and clauses, reducing up to 360,000 h annually, consuming significant human resources and enhancing the precision and scalability of operations, setting a new standard for efficiency in financial reporting. Another compelling example comes from Deloitte, a powerhouse in audit and assurance services which employed ACL Analytics, a sophisticated data analytics software, to elevate its audit processes. This tool harnesses the power of AI and machine learning to sift through vast volumes of financial data, identifying anomalies, trends, and patterns that warrant further scrutiny. By integrating neural networks, auditors can concentrate their efforts on high-risk areas, thereby improving the quality and depth of their audits and

providing clients with valuable, data-driven recommendations, fostering more informed decision-making and strategic planning.

3. Materials and Methods

This research paper uses a comprehensive foundational research methodology to unlock business value by integrating AI-based decision-making with digital transformation in financial reporting systems. The logical process described in this methodology serves as a step-by-step road map, guiding the investigation into the intersection between artificial intelligence (AI) and digital transformation in the economy in the context of the digitalization of financial reporting. The main objective is to develop a theoretical framework that improves decision-making processes by assimilating advanced artificial intelligence technologies into existing financial reporting systems. Through this research we went through systematic stages, starting with the identification of the research problem, followed by an extensive literature review to gather relevant information on artificial intelligence, digital transformation, and current financial reporting systems. The theoretical foundation was then mapped by synthesizing existing knowledge and identifying gaps in current understanding. Subsequently, the authors formulated a research design to guide the empirical investigation.

Integrating AI-based decision-making with digital transformation into financial reporting systems requires careful consideration of technological, organizational, and ethical dimensions. The methodology addresses these issues by incorporating a multidisciplinary approach, ensuring a holistic understanding of the challenges and opportunities associated with implementing artificial intelligence in financial reporting.

By navigating through this methodical process, the research aims to provide valuable information to both academia, business, and industry. The expected outcome is a solid theoretical framework that not only clarifies the synergies between artificial intelligence and digital transformation but also provides practical guidance for organizations seeking to improve their financial reporting decision-making capabilities.

This research aims to bridge the gap between theoretical advances and practical implications, thereby unlocking the untapped business value inherent in integrating AI-driven decision-making with digital transformation in financial reporting systems.

4. Digital Transformation Tools for the Decision-Making Process

Digital transformation tools for financial reporting can significantly support the decision-making process within organizations, developing strategies that are resilient to different economic conditions. They play a crucial role in supporting decision-making processes in financial reporting by providing real-time data, advanced analytics, automation, and collaborative features. These tools contribute to more informed, strategic, and data-driven decision-making within organizations. The integration of advanced technologies and digital tools into financial reporting systems offers several benefits that enhance decision-making capabilities. Decision-makers can access up-to-date information on key financial metrics, performance indicators, and market trends, allowing for more informed and timely decision-making [17,18].

Advanced analytics and data visualization tools help transform complex financial data into easily understandable visual representations. This aids decision-makers in identifying patterns, trends, and anomalies, facilitating data-driven decision-making. Digital tools often incorporate forecasting and predictive analytics capabilities. By analyzing historical data and identifying patterns, these tools can assist decision-makers in making more accurate predictions about future financial trends and outcomes. Incorporating artificial intelligence and machine learning (ML) algorithms enhances the capabilities of financial reporting systems. These technologies can provide intelligent insights, identify opportunities, and support decision-making by analyzing large datasets and facilitating collaboration and communication among different stakeholders involved in the decision-making process [14,15].

Decision-makers can assess the financial implications of various decisions, ensuring alignment with regulatory requirements and risk mitigation strategies and enabling scenario planning by allowing decision-makers to model different financial scenarios and assess their potential impact. Analysis of strategic interactions among multiple decision-makers, often in competitive situations, can be employed to support managers and Shareholders in the decision-making process. In business contexts, it can be used to model and analyze decision-making scenarios involving multiple stakeholders, such as competitors, suppliers, and customers [19,20].

Game theory and machine learning provide strategic insights for risk assessment by analyzing diverse sets of data and identifying potential trends or anomalies and by helping managers and Shareholders anticipate the actions and reactions of others [18]. They aid in the making of decisions that consider the likely responses of competitors and collaborators, leading to more informed strategies. Neural networks, a subset of machine learning and AI, are adept at pattern recognition and prediction. In decision-making, neural networks can analyze large datasets, identify patterns, and make predictions based on historical and real-time information. This capability is valuable for managers and Shareholders when evaluating various decision options and their associated risks.

This conceptual map (Figure 2), provides a visual representation of the interconnected concepts, illustrating how game theory and neural networks, when integrated into decision support systems, contribute to strategic decision-making processes and enhance overall decision quality for stakeholders [21]. The combination of game theory and neural networks can be particularly powerful. Neural networks can analyze data to inform decision-makers about the likely outcomes of different strategies, while game theory can help model the strategic interactions and competitive dynamics among stakeholders. By leveraging neural networks to analyze historical data and predict outcomes, decision-makers can use game theory to optimize their strategies based on a deeper understanding of the competitive landscape. AI, including game theory and neural networks, can be integrated into decision-support systems that provide actionable insights to managers and Shareholders.

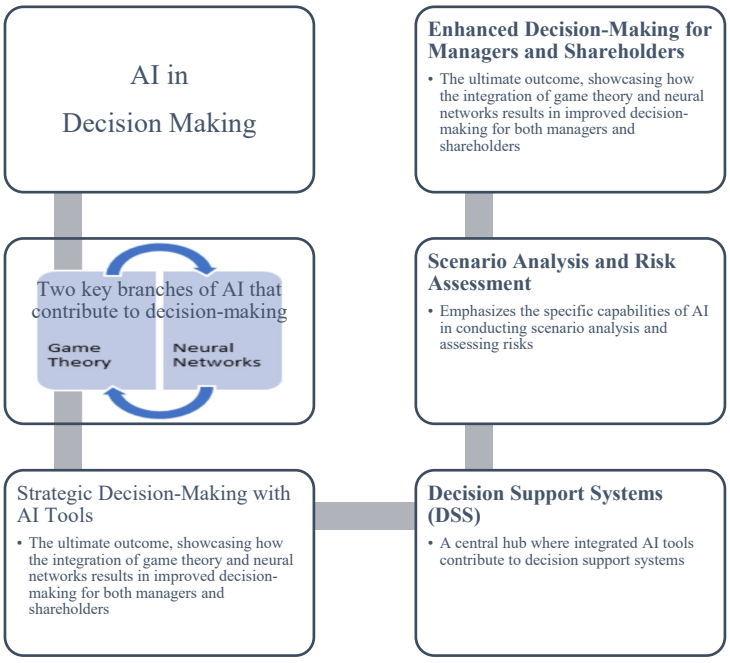


Figure 2. Conceptual map of AI integrated into decision support systems.

These systems leverage advanced analytics to enhance decision-making processes. Managers can use AI tools in the decision-making process to conduct scenario analysis,

assessing the potential outcomes of different decisions under various conditions. This aids in the making of decisions that are more robust and adaptable to different circumstances

4.1. Developing a Neural Network for Integrating AI in Decision-Making Processes

Developing a neural network for integrating AI in financial reporting involves designing a model that can analyze and interpret financial data from balance sheets. The choice of a three-layer neural network for accounting and financial reporting tasks is not an inherent rule, but rather a commonly used architecture that has shown effectiveness in various applications. A three-layer architecture is relatively simple and easier to interpret when we compare it to deeper counterparts. This architecture has fewer parameters and layers, which ostensibly makes it more manageable. However, this should not obscure the reality that even a three-layer neural network can function as a complex black box. The inherent non-linear transformations and the interactions between layers make it challenging to decipher how specific inputs are translated into outputs. In financial reporting, where transparency and accountability are paramount, the opaque nature of neural networks, regardless of their depth, can lead to significant interpretability challenges. Because interpretability is often crucial, and as stakeholders need to understand and trust the results of the model, this complexity is not just a theoretical concern but a practical one that impacts trust and reliability in real-world applications.

Three-layer neural networks are notably powerful due to their ability to approximate a wide range of functions. This capability is rooted in Kolmogorov's superposition theorem, which asserts that a three-layer neural network can represent any multivariate function, whether continuous or discontinuous. This makes such networks versatile for various applications, including financial reporting systems where capturing complex, non-linear relationships within data is crucial [22,23]. One of the primary advantages of a three-layer neural network is its relative simplicity and interpretability compared to deeper networks. While it can be challenging to interpret the decision-making process of very deep neural networks, three-layer models strike a balance by being complex enough to capture intricate patterns while remaining simpler to analyze and debug. This interpretability is particularly important in financial contexts where transparency and accountability are paramount [24,25].

Simpler models are also less prone to overfitting and can be more robust, especially when dealing with limited data. For many accounting and financial reporting tasks, a moderate number of hidden layers can effectively capture the underlying patterns and relationships in the data.

We designed the mathematical model for a neural network with 31 input units, 16 hidden units, and 2 output units as follows:

$$x1, x2, ..., x31 \text{tobetheinputfeatures}$$

$$h1, h2, ..., h16 \text{ tobethehiddenlayerunits}$$

$$y1, y2 \text{ tobetheoutputlayerunits}$$

The mathematical model of the neural network can be expressed as follows:

Hidden layer computation:

$$hj = \text{ReLU}\left(\sum i = 131 wij(1)xi + bj(1)\right), \text{for } j = 1, 2, \ldots, 16$$

where $\text{ReLU}(a) = \max(0, a)$ is the Rectified Linear Unit activation function.

Output layer computation:

$$yk = \text{Softmax}\left(\sum j = 116 wjk(2)hj + bk(2)\right), \text{for } k = 1, 2$$

were

$$\text{Softmax}(a)i = \sum j = \frac{eai}{\sum 1 eai} \text{ istheSoftMaxactivationfunction.}$$

The parameters of the model (weights and biases) are learned during the training process to minimize a specific loss function, typically associated with the task at hand (e.g., classification or regression). The training involves adjusting the weights and biases using optimization algorithms such as stochastic gradient descent (SGD).

The model is trained by iteratively performing forward propagation, calculating the loss, and then using backpropagation to update the model parameters. The training process continues until the model converges or until a predetermined number of epochs is achieved. Once trained, the neural network can be used to make predictions on new data by performing forward propagation with the learned parameters. This mathematical approach forms the basis of training using a three-layer neural network for financial reporting tasks, connecting the input layer with the balance sheet structure to the hidden layer, and ultimately predicting financial KPIs in the output layer. Adjustments can be made based on the specific characteristics of your data and task requirements.

In financial settings where data can be limited and computational resources may be constrained, a three-layer architecture strikes a balance between complexity and efficiency. Extremely deep networks may suffer from vanishing or exploding gradient problems during training, making it challenging for the model to learn effectively. A three-layer network is less prone to these issues, especially when using activation functions like ReLU (Rectified Linear Unit) that mitigate the vanishing gradient problem [26].

The input layer should be designed to capture relevant information from the balance sheet. As seen in Figure 3. And further explained in Appendix A. Each node in the input layer corresponds to a specific feature or variable from the balance sheet, such as assets, liabilities, equity, taxes, depreciation, net revenue, etc. Including features related to the balance sheet structure allows the neural network to learn patterns and relationships within the financial data. This information is crucial for understanding the financial health and position of the economic entity. Although this is a proposal model for an economic entity, the effectiveness of our neural network depends on the quality and relevance of the accounting provided. It should be ensured that the selected features in the input layer are meaningful and that there is sufficient data to train the model effectively.

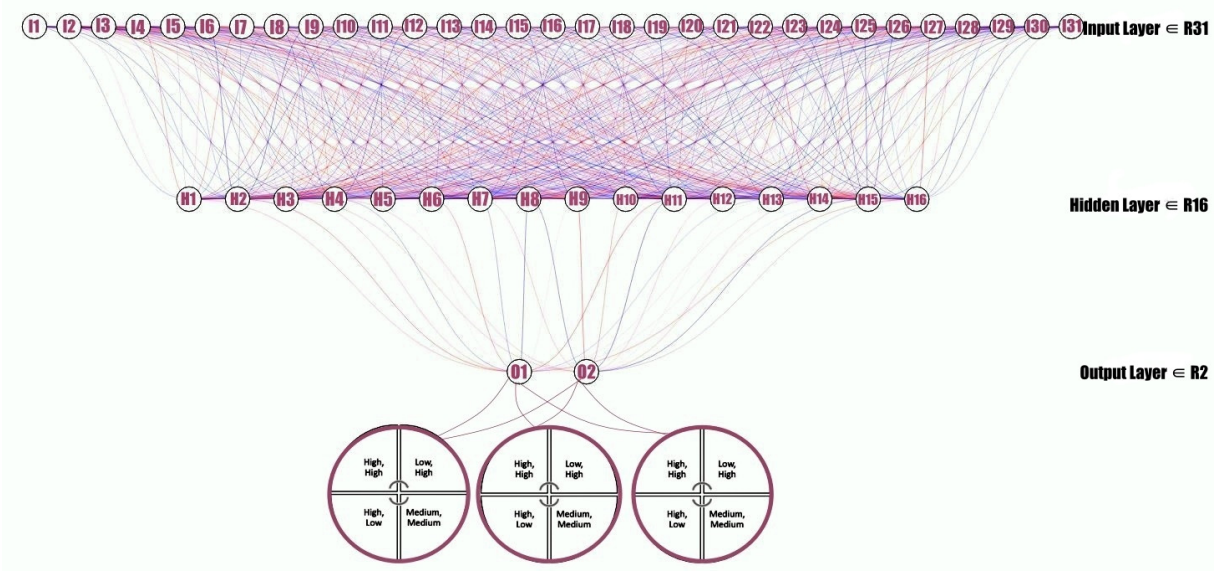


Figure 3. Three-layer neural network for accounting and financial reporting tasks.

The hidden layer is where the neural network learns complex representations and patterns from the input features. By connecting the input layer to the hidden layer, the network can extract meaningful insights and relationships. If the economic entity's goal is to increase business value, it makes sense to correlate the hidden layer with financial KPIs. Financial KPIs often serve as performance indicators that directly impact the overall

value and success of a business. The hidden layer can learn to abstract and combine features from the input layer to identify patterns associated with successful financial outcomes or increased business value. The choice of the two output layers depends on the specific objectives and how the company defines and relates to business value, whether by profitability or revenue growth. The exact KPIs may vary depending on the industry, company goals, and the nature of the business that integrates the neural network AI tool. One of the key indicators of a successful business is its ability to generate increasing revenue over time. The first outcome layer could represent the predicted revenue growth. The neural network would then be trained to learn patterns in the input data that correlate with higher revenue.

The mathematical model would have two output nodes, each corresponding to one of these outcomes. For binary classification, the company might use a sigmoid activation function in the output layer, and for regression tasks, the company might use linear activation [17].

$$(1)\ Y_1 = \sigma(H \cdot {W_1}^{(2)} + {b_1}^{(2)})$$

$$(2)\ Y_2 = \sigma(H \cdot {W_2}^{(2)} + {b_2}^{(2)})$$

where σ is the sigmoid activation function, and Y_1 and Y_2 represent the predicted values for revenue growth and profitability, respectively.

The third output node can be included if the economic entity might be able to account for negative financial KPIs that they want to minimize like operating costs or the expense ratio.

$$(3)\ Y_3 = \sigma(H \cdot {W_3}^{(2)} + {b_3}^{(2)})$$

4.2. Using Game Theory AI Tools in the Decision-Making Process

Game theory is powerful in the era of digitalization [20] and AI because it provides a structured approach to decision-making in complex, dynamic, and uncertain environments. It helps decision-makers navigate strategic interactions, allocate resources efficiently, and adapt to the rapidly changing landscape of technology and business. As we previously mentioned, we encourage economic entities to apply game theory to their business processes to unlock business value through AI-driven decision-making and digital transformation in financial reporting systems and create a payoff matrix that represents the interactions between different decision-makers or stakeholders. In the context of our research, these decision-makers could include the Company Management, Shareholders, AI system, and other relevant entities.

We assume there are two key decision-makers: Company Management (CM) and Shareholders (SH), and there are two strategic choices for each: implement AI-driven financial reporting (AI) or stick to traditional reporting (TR). The payoffs are expressed in terms of business value or utility for each combination of choices.

Explanation of the Payoffs

- High, High: If both Company Management and Shareholders choose to implement AI-driven financial reporting, they both receive high business value or utility.
- Low, High: If Company Management opts for AI while Shareholders stick to traditional reporting, Company Management might experience low value (due to the implementation costs, for example), while Shareholders receive high value (as they may prefer the familiar traditional reporting).
- High, Low: If Company Management sticks to traditional reporting while Shareholders prefer AI-driven reporting, Company Management might achieve high value (as they avoid implementation costs), but Shareholders receive low value (as they desire the benefits of AI-driven reporting).
- Medium, Medium: If both Company Management and Shareholders stick to traditional reporting, they both achieve a medium level of business value or utility.

We can elaborate on the payoffs in terms of increasing value for both Company Management and Shareholders based on two strategic choices: implement AI-driven financial reporting or stick to traditional reporting. The values represent the perceived business value or utility, with higher values indicating higher perceived value. High, medium, and low are qualitative representations of the perceived value, and the actual numerical values would depend on the specific context and preferences of stakeholders. The payoffs are described in Table 1, and the actual values would depend on factors such as the industry, the specific goals and preferences of stakeholders, and the perceived benefits and drawbacks of AI-driven reporting versus traditional reporting in the given context. This matrix provides a structured way to understand how the decisions of each party influence the perceived value for both Company Management and Shareholders:

Table 1. Payoff matrix.

		Shareholders	
Company Management		AI	TR
	AI	High, High	Low, High
	TR	High, Low	Medium, Medium

- High, High (AI):
  - Company Management (CM): High value, as AI-driven financial reporting is expected to enhance efficiency, accuracy, and strategic decision-making, leading to increased overall business performance.
  - Shareholders (SH): High value, as they benefit from improved transparency, better-informed decision-making by management, and potentially increased profits.
- Low, High (TR):
  - Company Management (CM): Low value, as sticking to traditional reporting may result in missed opportunities for efficiency gains, strategic insights, and cost savings offered by AI-driven reporting.
  - Shareholders (SH): High value, as they may prefer the familiarity and stability of traditional reporting, potentially perceiving less risk or disruption to their investments.
- High, Low (AI):
  - Company Management (CM): High value, as the implementation of AI-driven reporting satisfies the management's objective of adopting innovative technologies and staying competitive.
  - Shareholders (SH): Low value, as they might be disappointed with the decision not to stick with traditional reporting, potentially perceiving higher risks or uncertainties.
- Medium, Medium (TR):
  - Company Management (CM): Medium value, as sticking to traditional reporting may provide stability but may not leverage the potential benefits offered by AI-driven reporting.
  - Shareholders (SH): Medium value, as they maintain a sense of stability but may miss out on potential improvements in decision-making and efficiency.

CFOs face both challenges and opportunities in the AI decision-making process within a given industry and regulatory environment. These can vary based on the specific circumstances of each organization. They must navigate complex data privacy regulations to ensure that the use of AI aligns with privacy laws, particularly in industries dealing with sensitive customer information. Another challenge that management faces soon is keeping up with evolving regulations related to AI and ensuring that AI systems comply with industry-specific laws and standards.

Game theory matrices can be tailored to various decision-making scenarios, and when applied to the context of Chief Financial Officers (CFOs) making decisions in the realm of AI, we propose four matrices that can be taken into consideration in the decision-making process: CFOs making AI-related decisions can use game theory matrices to navigate four key scenarios: early vs. late AI adoption, external vendor vs. in-house AI development, data sharing collaboration vs. data protection, and proactive vs. reactive regulatory compliance, balancing benefits, costs, and risks for competitive advantage and innovation.

4.3. Ethical and Privacy Issues in AI-Driven Financial Reporting

In this relatively new AI-driven financial reporting environment, data privacy concerns are paramount. The extensive data requirements of AI systems often encompass sensitive financial and personal information, raising the specter of unauthorized access and data breaches [27]. The very foundation of our work relies on the integrity and confidentiality of the data we handle. As we integrate AI, we must be vigilant in protecting these data through robust encryption methods, ensuring that sensitive information remains secure both in transit and at rest. Stakeholders must be fully informed about how their data are collected, stored, and utilized. This transparency extends to obtaining explicit consent from individuals whose data are being used. By implementing stringent data governance policies, we can delineate clear access controls and ensure that data are only accessible to authorized personnel under appropriate circumstances. Accountability and transparency in AI systems are critical and it is imperative that app developers establish clear accountability frameworks that define the responsibilities of AI users, operators, and decision-makers. When errors occur, there must be a transparent process for identifying and addressing the source of the problem. Implementing explainable AI techniques can significantly aid this process. These techniques allow us to understand and articulate how AI systems make decisions, thereby fostering trust and accountability among stakeholders [28].

With the widespread adoption of international accounting standards, regulatory compliance is an ever-present concern in our profession. AI applications in financial reporting must adhere to a myriad of regulatory requirements, from financial regulations to data protection laws. Engaging in ongoing dialogue with regulatory bodies can provide valuable guidance and help ensure that our AI systems are compliant with current laws. Regular compliance audits are essential to verify adherence and address any gaps in our processes.

Security risks are an inherent aspect of integrating AI into financial reporting. AI systems can be targeted by cyberattacks, potentially leading to data breaches or the manipulation of financial information [29]. To mitigate these risks, we must implement advanced cybersecurity measures and develop comprehensive incident response plans. Regular security assessments will help identify and address vulnerabilities, ensuring the integrity and reliability of our AI systems. In real-life situations, several advanced cybersecurity measures have proven effective in safeguarding AI-driven financial reporting systems, such as multi-factor authentication that significantly enhances security by requiring users to provide two or more verification factors to gain access to a system, end-to-end encryption that ensures data are encrypted from the moment they are created until they are received and decrypted by the intended recipient, or conducting regular security audits and penetration testing that allow organizations to identify and address vulnerabilities in their systems proactively [29].

5. Results and Discussion

Combining a neural network model with game theory, as represented in a matrix, can be a powerful approach, and it is often referred to as game-theoretic machine learning and is used for decision support in companies. The neural network can be used to predict financial KPIs or other relevant outcomes, and the game theory matrix can help analyze the strategic interactions between different entities or stakeholders. It aims to understand how different agents or players, each with their own objectives, make decisions in a competitive or

cooperative environment. We use game theory as a branch of mathematics and economics to monitor the interactions between rational decision-makers, often referred to in the literature as players, in situations where the outcome of one player's decision depends on the decisions of others, and we propose to combine machine learning techniques and integrate them with game theory to model and predict the behavior of players who can gain from positive business value creation. In our game theory, the players represent the Shareholders and the management team, and strategies include increased market capitalization by increasing revenues or profitability and payoffs represented in the matrix. Strategies proposed by the research are the choices available to each player such as choosing the digitalization and incorporation of AI path or sticking to traditional methods of accounting and reporting, and payoffs represent the outcomes associated with specific combinations of strategies chosen by all players. To apply the strategies, predictive models, such as the neural networks with tree layers, can be employed to estimate the likely choices or actions of players based on historical accounting data extracted from the balance sheet and profit and loss data or other relevant features.

Given the specifications of our neural network and the structure of the payoff matrix from game theory, we suggest naming the theoretical model "Decision Harbor AI" a name that reflects the integration of artificial intelligence, financial decision-making, and strategic analysis and symbolizes a safe and informed harbor for decision-making aided by artificial intelligence. Game-theoretic machine learning provides a framework for understanding and predicting strategic interactions [30]. By integrating machine learning models with game theory concepts, we predict it will become possible to gain insights into decision-making processes in complex, dynamic environments. This approach can be valuable for making informed decisions in competitive or cooperative scenarios where multiple stakeholders are involved [31].

In our scenario, we have a game theory matrix with two players: Company Management and Shareholders and the matrix represents the possible outcomes based on the decisions made by these players. We can train the neural network to predict the financial KPIs or outcomes of interest, such as revenue growth, profitability, and cost efficiency as seen in Figure 4. After we trained the neural network to predict positive performance indicators, we used the trained neural network to make predictions for each player (Company Management, AI, and TR) based on the input features and the KPI in the hidden layer.

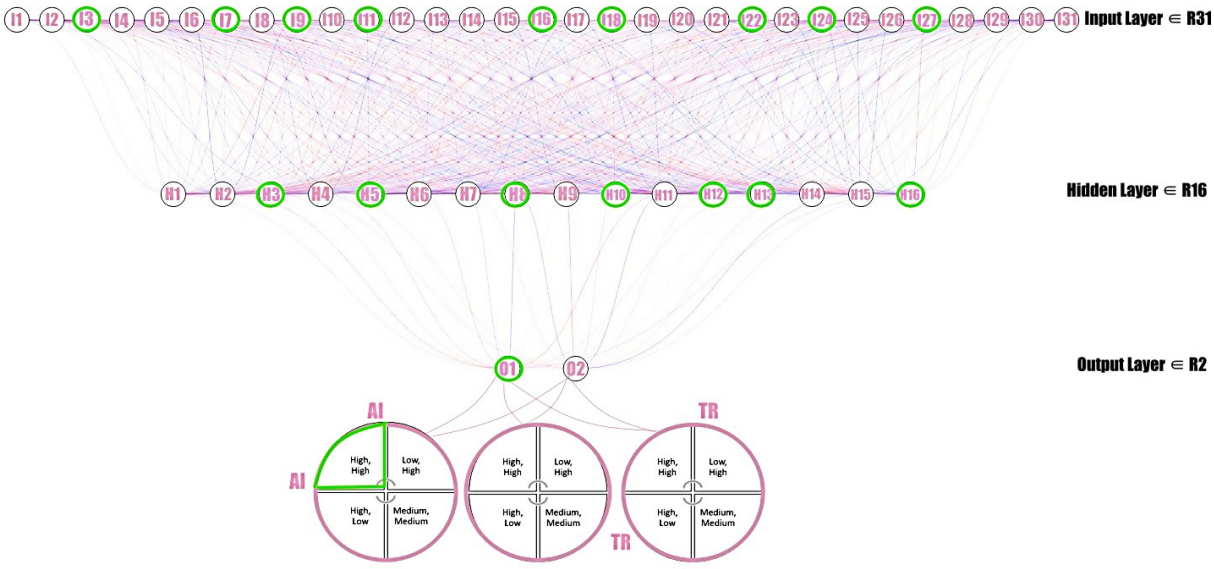


Figure 4. Decision Harbor AI theoretical model—scenario 1.

In bad financial years, the distribution of data may shift. The patterns and relationships between input features and financial outcomes may change, leading to a mismatch between the training and testing data. The neural network, trained on historical data, may struggle to generalize to these new conditions. The model's predictive performance may degrade during bad financial years if the patterns observed during training no longer hold. The model might make inaccurate predictions, especially if it has not encountered similar scenarios in the training data (Figure 5).

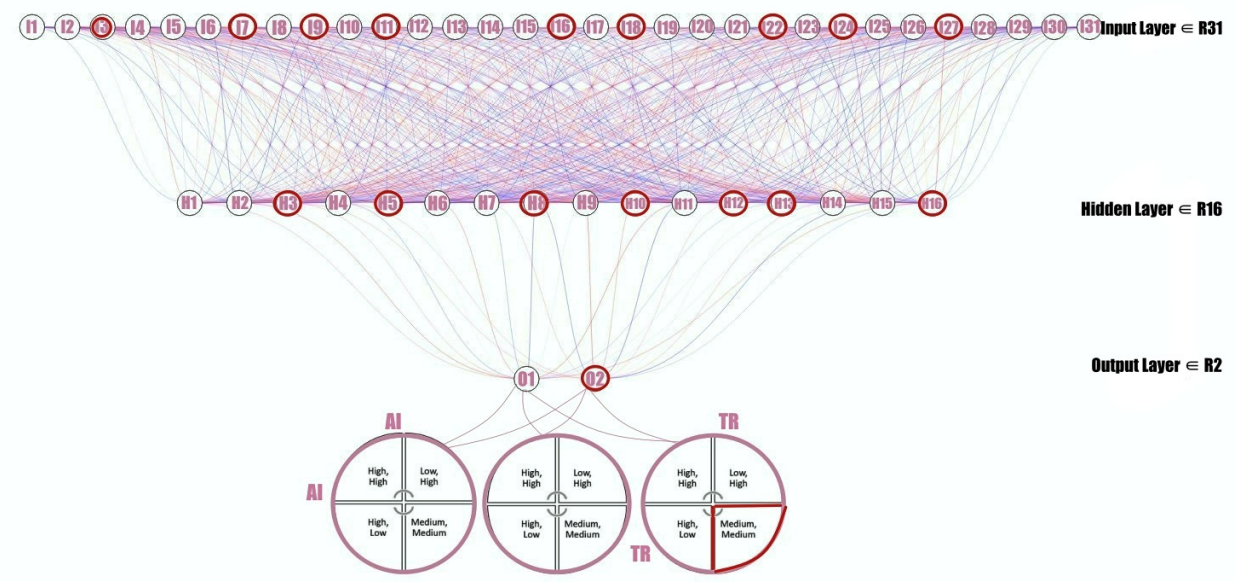


Figure 5. Decision Harbor AI theoretical model—scenario 2.

We suggest that the company implements a strategy for continuous model training to adapt to evolving financial conditions. The model should be regularly updated with new data, especially during periods of financial turbulence, to ensure it remains relevant. For the model to lead to good financial decisions, the management must conduct a scenario analysis to assess the model's performance under various financial conditions. This may involve testing the model against simulated scenarios that represent different economic states, including good and bad financial years. If the neural network performs well during good financial years, it provides validation of the model's robustness. It suggests that the model can adapt to different financial conditions and continues to make accurate predictions when conditions are favorable. Good financial years can positively reinforce the use of the model for decision-making. Reliable predictions in our company during these periods can contribute to effective strategic planning, resource allocation, and other decisions that enhance business value. Decision-makers may gain increased confidence in the model's predictions during good financial years. This confidence can lead to more reliance on the model for guiding decisions.

The Deming Cycle (PDCA) may be used as an iterative four-step management method for continuous improvement. Applying the PDCA cycle to the continuous training and scenario analysis of our neural network model in the context of evolving financial conditions is a sensible and effective approach. We must monitor the training process and assess whether the model's performance improves or remains stable, execute the scenario analysis using the prepared simulation scripts or tools, and assess the model's accuracy, sensitivity, and specificity across different scenarios to identify any patterns of underperformance or overfitting.

If the scenario analysis of our neural network reveals shortcomings, we must consider adjusting the model or the scenario generation process. This may involve redefining features, introducing new features, or modifying the simulation approach. The theoretical

model can incorporate any necessary improvements to enhance the model's ability to handle diverse financial conditions. We can apply stress testing to our model [23], like how banks conduct stress tests to assess the resilience of their financial systems under adverse conditions. In the context of our neural network model and game theory matrix, stress testing can help us understand how well the model performs and how robust decision-making is in the face of extreme or unexpected financial scenarios. Deep learning can be applied to the input layer in our network in the context of dynamic balance sheet stress testing [31,32]. If the balance sheet that generates financial information for the input layer data has a temporal dimension like quarterly or yearly, we can consider recurrent neural networks or Long Short-Term Memory networks (LSTMs) to capture temporal patterns and dependencies in the data. If our input data include diverse sources like financial statements, market data, and economic indicators, we suggest applying architectures that support multimodal data integration. This allows the model to learn from different types of information simultaneously. Overall, it is important to tailor the deep learning approach to the specific requirements and characteristics of our financial data and stress testing objectives. Our theoretical model DecisionHarborAI can be implemented in public organizations as well. The application of AI tools, like this theoretical two-layer model, in the public sector has the potential to bring about significant benefits like assisting in analyzing the impact of different policies, helping policymakers understand potential outcomes and make informed decisions, providing insights into optimizing resource allocation to achieve desired outcomes, evaluating the effectiveness of public programs and initiatives, and enhancing citizen engagement by providing data-driven insights into decision-making processes, fostering transparency and trust between the public and government [33].

To effectively implement such a model, several practical recommendations emerge. Regular updates and retraining of the neural network with new data are crucial to maintaining accuracy and relevance.

The first step is providing a picture or a PDF file of a balance sheet to our API. This will be conducted in a mobile app or a web application. As soon as a picture or PDF is received, it is converted to a TXT file. A parser takes the TXT gained from the OCR and converts it into structured JSON using machine learning. The JSON is then returned as the output from the API. From here, data from the balance sheet can be processed further. The comprehensive dataset is then used to train a neural network, capable of predicting key financial performance indicators such as revenue growth, profitability, and cost efficiency. The DecisionHarborAI model goes a step further by incorporating a game theory framework. This allows for a nuanced analysis of strategic interactions between stakeholders, such as management and Shareholders. By understanding these dynamics, the institution can align its strategic decisions with the interests of all parties involved, thereby maximizing business value.

To ensure robustness, continuous scenario analysis and stress testing are conducted. This practice not only validates the model's predictions under varying conditions but also prepares the institution for potential economic fluctuations. The outcomes are profound: enhanced decision-making capabilities, significant efficiency gains through automation, and a strategic alignment that drives overall business performance.

Limitations must also be considered when implementing AI-driven financial reporting models such as DecisionHarborAI. One of the most significant limitations is the quality of the data fed into the AI models. AI systems require vast amounts of high-quality data to function optimally. However, financial data can often be incomplete, inconsistent, or noisy. Inaccurate or poor-quality data can lead to erroneous predictions and flawed decision-making processes. Organizations must invest considerable effort in data cleansing and validation to ensure that the data used for training and operational purposes is accurate and reliable. This can be resource-intensive and may not always be feasible, particularly for smaller institutions with limited budgets. Like all AI models, DecisionHarborAI will be trained on historical data and may perform well under conditions similar to those present in the training dataset. The ability of our theoretical model to adapt to economic shifts,

regulatory changes, and unforeseen global events such as financial crises or pandemics is a critical concern [34]. The DecisionHarborAI model may fail to generalize effectively to new, unseen scenarios, leading to performance degradation. Continuous model retraining and validation will be necessary to ensure that DecisionHarborAI remains relevant and accurate, but this can be challenging to manage and requires ongoing investment.

6. Conclusions

When implementing a neural network for financial accounting, it is important to consider factors such as data quality, model interpretability, and the specific regulatory environment. The model that this research proposes offers an integrated approach to decision support by combining the predictive capabilities of neural networks with the strategic insights provided by game theory. This integration aims to enhance decision-making processes in complex and dynamic environments and shifts the model's focus to predicting financial indicators and supporting financial decision-making. The neural network component of the new AI tool is designed to provide financial predictions and insights based on historical data. This includes forecasting key financial indicators such as revenue growth, profitability, and cost efficiency. Neural networks can be a valuable tool for calibrating outcomes in the financial accounting environment, and have demonstrated effectiveness in processing complex financial data, identifying patterns, and making predictions. The inclusion of a game theory payoff matrix in this new AI decision-making tool introduces a strategic framework for analyzing interactions among decision-makers. This allows for the consideration of strategic choices and outcomes in a more dynamic and competitive context. AI digital transformation tools have the potential to significantly enhance financial decision-making processes and overall strategic management. These tools can help organizations leverage data-driven insights and automate routine tasks, potentially improving efficiency and decision accuracy. However, it is important to recognize that AI technologies also come with challenges and disadvantages, such as data privacy concerns, the risk of algorithmic biases, and the need for substantial computational resources. Additionally, the complexity of AI models can make them difficult to interpret and trust, which may limit their practical application in certain contexts. Therefore, while AI offers promising advancements, it is essential to approach its integration with a balanced perspective, acknowledging both its benefits and its limitations. All future developing decision-making assistance tools must continuously learn and update their predictions, ensuring relevance and accuracy over time. The model that this paper suggests must be equipped with the capability to undergo stress testing and scenario analysis, allowing it to evaluate its performance under various financial conditions, including both favorable and adverse scenarios, contributing to its robustness.

Neural network payoff matrix fusion represents a comprehensive and adaptive decision support tool that leverages the strengths of neural networks and game theory to provide valuable insights for financial decision-making in both private and public sectors and provides tangible value to businesses. The continuous training, scenario analysis, and strategic framework contribute to its potential effectiveness in dynamic and uncertain environments. The concept of integrating AI-driven decision-making is directly reflected in the conclusions of this paper. The digital tools mixed combine the predictive capabilities of neural networks with a game theory framework, emphasizing the integration of advanced AI techniques for more informed decision-making. The model's success will depend on thorough testing, collaboration with domain experts, and ongoing refinements based on real-world feedback.

Digital transformation tools can significantly aid the financial decision-making process and overall strategic management in a company by analyzing historical financial data to make accurate predictions about future trends and outcomes. These tools can assess and predict potential risks by analyzing various data sources. This enables organizations to proactively identify and mitigate risks in financial decision-making and to automate repetitive and routine tasks, such as data entry, reconciliation, and reporting.

AI digital transformation tools play a crucial role in enhancing financial decision-making processes and overall strategic management. These tools empower organizations to leverage data-driven insights, automate routine tasks, and navigate the complexities of the modern business landscape more effectively.

Author Contributions: Conceptualization, A.E.A. and A.E.D.; A.E.D. and L.I.: formal analysis, A.E.A.: investigation, L.I.: resources, A.E.D.: writing—original draft preparation, A.E.A.: writing—review and editing, A.E.D.: visualization, L.I.: supervision, A.E.A.: project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data underlying the results are available as part of the article and no additional source data are required.

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

Appendix A

Table A1. Description of the layers in the proposed neural model—Decision Harbor AI.

Input Layer $\in \mathbf{R^{31}}$ **I**	**Hidden Layer** $\in \mathbf{R^{16}}$ **H**	**Output Layer** $\in \mathbf{R^{2}}$ **O**
Net revenue	Gross revenue	Revenue Growth
Net Income		
Taxes	Gross income	
Interest expenses	EBITDA	
Depreciation		
Amortization	Profit margins	
Cost of goods sold		
Operating expenses	Cash Flow from operations	
Operating income		
Change in working capital	Free cash flow	
Operating Cahs flow		
Capital expenditure	ROE	
Shareholders' equity		
Average total assets	ROA	
Total liabilities		
Current assets	Debt to equity ration	
Current liabilities		
Cash	Current ratio	Profitability
Cash equivalents		
Net accounts receivable	Quick ratio	
Prepaid expenses		
Share price	Price to earnings ratio	
Earnings per share		
Preferred equity	Book value per share	
Dividend per share		
Value per share	Dividend yield	
Total shares		
Revenue growth	Market capitalization	
Earnings		
Godwill	Growth metrics	
Fixed assets		

References

1. Imoniana, J.O.; Reginato, L.; Junior, E.B.C.; Benetti, C. Stakeholders' perceptions on the adoption of International Financial Reporting Standards by small and medium-sized enterprises: A multidimensional analysis. *Int. J. Bus. Emerg. Mark.* **2025**, *1*. [CrossRef]
2. Radicic, D.; Petković, S. Impact of digitalization on technological innovations in small and medium-sized enterprises (SMEs). *Technol. Forecast. Soc. Chang.* **2023**, *191*, 122474. [CrossRef]
3. Saura, J.R.; Ribeiro-Soriano, D.; Palacios-Marqués, D. Assessing behavioral data science privacy issues in government artificial intelligence deployment. *Gov. Inf. Q.* **2022**, *39*, 101679. [CrossRef]
4. Faccia, A.; Al Naqbi, M.Y.K.; Lootah, S.A. Integrated Cloud Financial Accounting Cycle. In Proceedings of the 2019 3rd International Conference on Cloud and Big Data Computing, Oxford, UK, 28–30 August 2019; ACM: New York, NY, USA, 2019; pp. 31–37. [CrossRef]
5. Allam, K.; Rodwal, A. AI-driven big data analytics: Unveiling insights for business advancement. *EPH-Int. J. Sci. Eng.* **2023**, *9*, 53–58. [CrossRef]
6. Mengu, D.; Luo, Y.; Rivenson, Y.; Ozcan, A. Analysis of Diffractive Optical Neural Networks and Their Integration with Electronic Neural Networks. *IEEE J. Sel. Top. Quantum Electron.* **2020**, *26*, 1–14. [CrossRef]
7. Shengelia, N.S.N.; Tsiklauri, Z.T.Z.; Rzepka, A.R.A.; Shengelia, R.S.R. The Impact of Financial Technologies on Digital Transformation of Accounting, Audit and Financial Reporting. *Economics* **2022**, *105*, 385–399. [CrossRef]
8. Pacurari, D.; Nechita, E. Some considerations on cloud accounting. In *Studies and Scientific Researches. Economics Edition*; Faculty of Economic Sciences, "Vasile Alecsandri" University of Bacau: Bacau, Romania, 2013. [CrossRef]
9. Li, M.; Yu, Y.; Li, X.; Zhao, J.L.; Zhao, D. Determinants of SMEs' Transformation Toward Cloud Services: Perspectives of Economic and Social Rationalities. *Pac. Asia J. Assoc. Inf. Syst.* **2019**, *11*, 65–87. [CrossRef]
10. Sorensen, M.; Yasuda, A. Stakeholder impact of private equity investments. In *Handbook of the Economics of Corporate Finance: Private Equity and Entrepreneurial Finance*; Elsevier: Amsterdam, The Netherlands, 2023; pp. 299–341. [CrossRef]
11. Naik, N.; Hameed, B.Z.; Shetty, D.K.; Swain, D.; Shah, M.; Paul, R.; Aggarwal, K.; Ibrahim, S.; Patil, V.; Smriti, K.; et al. Legal and Ethical Consideration in Artificial Intelligence in Healthcare: Who Takes Responsibility? *Front. Surg.* **2022**, *9*, 862322. [CrossRef]
12. Vo-Thanh, T.; Zaman, M.; Hasan, R.; Akter, S.; Dang-Van, T. The service digitalization in fine-dining restaurants: A cost-benefit perspective. *Int. J. Contemp. Hosp. Manag.* **2022**, *34*, 3502–3524. [CrossRef]
13. Agustí-Juan, I.; Glass, J.; Pawar, V. A Balanced Scorecard for Assessing Automation in Construction. In Proceedings of the Creative Construction Conference 2019, Budapest, Hungary, 29 June–2 July 2019; Budapest University of Technology and Economics: Budapest, Hungary, 2019; pp. 155–163. [CrossRef]
14. Wang, Z.; Wang, Q.; Nie, Z.; Li, B. Corporate financial distress prediction based on controlling shareholder's equity pledge. *Appl. Econ. Lett.* **2022**, *29*, 1365–1368. [CrossRef]
15. Du, G.; Elston, F. RETRACTED ARTICLE: Financial risk assessment to improve the accuracy of financial prediction in the internet financial industry using data analytics models. *Oper. Manag. Res.* **2022**, *15*, 925–940. [CrossRef]
16. Song, Z.; Wu, S. Post financial forecasting game theory and decision making. *Financ. Res. Lett.* **2023**, *58*, 104288. [CrossRef]
17. Murali, M.; Kanmani, S. Modelling game theory applications using network elements to ensure Nash equilibrium through Price of Anarchy and Price of Stability. *Int. J. Syst. Syst. Eng.* **2024**, *14*. [CrossRef]
18. Hamidoğlu, A. A game-theoretical approach on the construction of a novel agri-food supply chain model supported by the government. *Expert. Syst. Appl.* **2024**, *237*, 121353. [CrossRef]
19. Zhuang, Y.T.; Wu, F.; Chen, C.; Pan, Y.H. Challenges and opportunities: From big data to knowledge in AI 2.0. *Front. Inf. Technol. Electron. Eng.* **2017**, *18*, 3–14. [CrossRef]
20. Mahmood, A.; Al Marzooqi, A.; El Khatib, M.; AlAmeemi, H. How Artificial Intelligence can leverage Project Management Information system (PMIS) and data driven decision making in project management. *Int. J. Bus. Anal. Secur. (IJBAS)* **2023**, *3*, 184–195. [CrossRef]
21. Chongcs, J.; Kathiarayan, V.; Chong, J.; Sin, C. The Role of Artificial Intelligence in Strategic Decision-Making Opportunities, Challenges, and Implications for Managers in the Digital Age. *Int. J. Manag. Commer. Innov.* **2023**, *11*, 73–79.
22. Laczkovich, M. A superposition theorem of Kolmogorov type for bounded continuous functions. *J. Approx. Theory* **2021**, *269*, 105609. [CrossRef]
23. Quraisy, A. Data Normality Using Kolmogorov-Smirnov and Shapiro-Wilk Tests. *J-HEST J. Health Educ. Econ. Sci. Technol.* **2020**, *3*, 7–11. [CrossRef]
24. Schmidt-Hieber, J. The Kolmogorov–Arnold representation theorem revisited. *Neural Netw.* **2021**, *137*, 119–126. [CrossRef]
25. Ismailov, V.E. A three layer neural network can represent any multivariate function. *J. Math. Anal. Appl.* **2023**, *523*, 127096. [CrossRef]
26. Elahi, M.; Afolaranmi, S.O.; Lastra, J.L.M.; Garcia, J.A.P. A comprehensive literature review of the applications of AI techniques through the lifecycle of industrial equipment. *Discov. Artif. Intell.* **2023**, *3*, 43. [CrossRef]
27. Habbal, A.; Ali, M.K.; Abuzaraida, M.A. Artificial Intelligence Trust, Risk and Security Management (AI TRiSM): Frameworks, applications, challenges and future research directions. *Expert. Syst. Appl.* **2024**, *240*, 122442. [CrossRef]
28. Cheng, L.; Liu, X. From principles to practices: The intertextual interaction between AI ethical and legal discourses. *Int. J. Leg. Discourse* **2023**, *8*, 31–52. [CrossRef]

29. Humphreys, D.; Koay, A.; Desmond, D.; Mealy, E. AI hype as a cyber security risk: The moral responsibility of implementing generative AI in business. *AI Ethics* **2024**. [CrossRef]
30. Cho, S.; Vasarhelyi, M.A.; Sun, T.; Zhang, C. Learning from Machine Learning in Accounting and Assurance. *J. Emerg. Technol. Account.* **2020**, *17*, 1–10. [CrossRef]
31. Ganaie, M.A.; Hu, M.; Malik, A.K.; Tanveer, M.; Suganthan, P.N. Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105151. [CrossRef]
32. Zhang, Z.; Cui, P.; Zhu, W. Deep Learning on Graphs: A Survey. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 249–270. [CrossRef]
33. Chandana Charitha, P.; Hemaraju, B. Impact of Artificial Intelligence on Decision-making in Organisations. *Int. J. Multidiscip. Res.* **2023**, *5*. [CrossRef]
34. Aldoseri, A.; Al-Khalifa, K.N.; Hamouda, A.M. Re-Thinking Data Strategy and Integration for Artificial Intelligence: Concepts, Opportunities, and Challenges. *Appl. Sci.* **2023**, *13*, 7082. [CrossRef]

electronics

MDPI

Article

Learning to Improve Operational Efficiency from Pose Error Estimation in Robotic Pollination

Jinlong Chen [1], Jun Xiao [1,*], Minghao Yang [2] and Hang Pan [3]

[1] School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541000, China; 7259@guet.edu.cn
[2] Research Center for Brain-Inspired Intelligence (BII), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China; mhyang@nlpr.ia.ac.cn
[3] Department of Computer Science, Changzhi University, Changzhi 046011, China; panhang@czc.edu.cn
* Correspondence: 22032303110@mails.guet.edu.cn

Abstract: Autonomous pollination robots have been widely discussed in recent years. However, the accurate estimation of flower poses in complex agricultural environments remains a challenge. To this end, this work proposes the implementation of a transformer-based architecture to learn the translational and rotational errors between the pollination robot's end effector and the target object with the aim of enhancing robotic pollination efficiency in cross-breeding tasks. The contributions are as follows: (1) We have developed a transformer architecture model, equipped with two feedforward neural networks that directly regress the translational and rotational errors between the robot's end effector and the pollination target. (2) Additionally, we have designed a regression loss function that is guided by the translational and rotational errors between the robot's end effector and the pollination targets. This enables the robot arm to rapidly and accurately identify the pollination target from the current position. (3) Furthermore, we have designed a strategy to readily acquire a substantial number of training samples from eye-in-hand observation, which can be utilized as inputs for the model. Meanwhile, the translational and rotational errors identified in the end-manipulator Cartesian coordinate system are designated as loss targets simultaneously. This helps to optimize the training of the model. We conducted experiments on a realistic robotic pollination system. The results demonstrate that the proposed method outperforms the state-of-the-art method, in terms of both accuracy and efficiency.

Keywords: pollination robot; transformer; offset errors

Citation: Chen, J.; Xiao, J.; Yang, M.; Pan, H. Learning to Improve Operational Efficiency from Pose Error Estimation in Robotic Pollination. *Electronics* **2024**, *13*, 3070. https://doi.org/10.3390/electronics13153070

Academic Editor: Dah-Jye Lee

Received: 24 June 2024
Revised: 26 July 2024
Accepted: 1 August 2024
Published: 2 August 2024

1. Introduction

Robot-assisted pollination has been a topic for over a decade [1]. It is becoming an increasingly important contributor to crop production in the agricultural industry, and it is providing solutions to pressing challenges, such as the decline in natural pollinators. Pollination robots are categorized into several types, to meet diverse agricultural needs. Autonomous robots, equipped with advanced vision and artificial intelligence, navigate and pollinate independently. Remotely controlled or semi-autonomous robots require human oversight, suitable for complex environments. Unmanned aerial vehicles (UAVs) efficiently pollinate large fields from above. These robotic systems are designed to mimic the pollination process traditionally performed by bees and other insects. Utilizing advanced sensors, vision systems, and precision manipulators, robotic pollinators identify and interact with flowers, depositing pollen with high accuracy and consistency.

Various studies have been discussed in the field of automated pollination with applications in the pollination of kiwifruit [2,3], bramble flowers [4], vanilla [5], and tomato flowers [6]. The pollination technique requires the physical transfer of pollen from the male to the pistils (the reproductive organs of the female flowers) [7]. To minimize pollen loss and ensure precise transfer to the stigmas, it is necessary to address the difficulties

of collecting pollen and the challenges of maintaining its activity [8]. Unfortunately, this requires considerable time and effort, which has not been carefully considered in the existing pollination approaches. The difficulties originate from the very small size of the pistils, as well as the challenge of accurately identifying their rotation and orientation.

To this end, this work proposes a methodology based on a transformer architecture that aims to enhance the operational efficiency of robotic pollination in cross-breeding tasks. The output layer of the model incorporates two separate feedforward neural networks, each designed to regress the translational and rotational discrepancies between the robot and the pollination target, respectively. Unlike traditional transformer loss functions, we have designed a novel loss function that is informed by the translational and rotational discrepancies between the robot's end effector and the pollination targets. This enables the robot arm to rapidly and accurately identify the pollination target from the current observation view. Meanwhile, we have also designed a strategy to collect large-scale training samples from eye-in-hand observation and to designate the translational and rotational errors identified within the end-manipulator Cartesian coordinate system as loss targets. The observed images in the current and target positions, combined with the translational and rotational errors between them, provide a large scale for training data, thereby enhancing the model training. Figure 1 shows the pollination robot used in this work.

The remainder of this work is organized as follows: the related works are outlined in Section 2; the detailed descriptions of the method are introduced in Section 3; Sections 4–6 present the experiments, discussion, and conclusion, respectively.

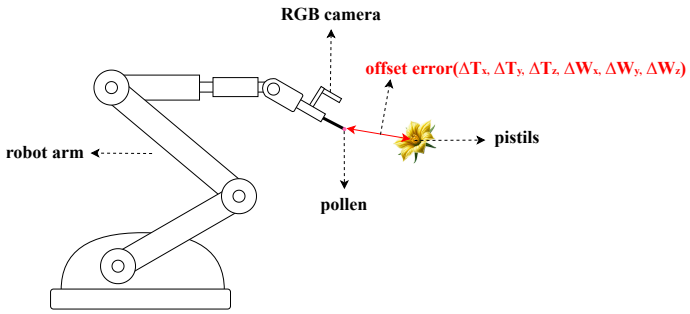


Figure 1. The pollination robot primarily consists of a UR5 robotic arm and a monocular RGB camera, configured in an eye-in-hand structure. At the end of the robotic arm, there is a pollination brush used for pollinating the pistils. The illustration depicts a robotic arm engaged in pollination, where the "offset error" marked in red represents the content that the model needs to learn. Here, ΔT_x, ΔT_y, and ΔT_z denote the components of the translational error along the X, Y, and Z axes, respectively, while ΔW_x, ΔW_y, and ΔW_z represent the components of the rotational vector, indicating the rotational errors.

2. Related Work

2.1. Pose Estimation

Achieving high precision in the end effector pose of a robotic arm poses significant challenges. To enhance the precision of the end effector pose, additional sensors, such as LiDAR or pressure sensors, are often employed, which can increase costs. Some methods are based on 6D object pose estimation, constructing templates to scan different positions in the input image, calculating similarity scores at each position, and obtaining the best match by comparing the similarity scores [9,10]. Others use feature-based methods, where CNN models directly regress the 3D coordinates of each pixel [11], or they describe the posterior density of a specific object pose through CNNs, comparing the observed image with the rendered image [12]. Additionally, some methods combine the advantages of template-based and feature-based approaches within a deep learning framework. The network integrates bottom-up pixel-level labeling with top-down object pose regression, predicting

the 3D position and 3D rotation of the object in a decoupled manner [13]. All these methods invariably use depth sensors to capture the depth information D of the target object. However, for scenarios where depth information cannot be captured, such as in the agricultural pollination robot discussed in this paper, the hollow regions of pollination flowers result in poor depth information acquisition. Moreover, the trained models incorporating depth information [11–13] do not directly regress and predict the 3D position information of the target object. The aforementioned template-based and feature-based models are trained on point clouds, but they still face challenges when handling non-convex polyhedra and non-manifold data. Existing point-cloud training methods are primarily suitable for regular shapes, such as convex polyhedra and manifolds. These methods are limited when dealing with the complex and irregular shapes of flowers in agricultural environments, which often present non-polyhedral and non-manifold structures. This limitation makes it difficult to directly apply traditional point-cloud training methods. Additionally, there is a discrepancy between point-cloud training in virtual environments and real-world applications, which can affect the model's performance in real-world scenarios.

By contrast to the pose-estimation techniques introduced earlier, which rely on additional sensors to capture depth-enhanced image data for constructing point-cloud training—a practice difficult to implement in non-convex and non-manifold agricultural environments—our method utilizes only conventional cameras, to acquire RGB images from the pollination robot's end effector. This approach enables seamless end-to-end training in real-world settings, thereby eliminating the discrepancies commonly found between point-cloud training in virtual environments and their practical application in the field.

2.2. Traditional and Deep Learning Methods for Improving Precision

To improve the precision of the pollination end effector, traditional methods can be realized through the design of the robotic arm. For example, LI K designed a robotic arm for kiwifruit pollination [14], achieving high precision by expressing the end effector velocity and joint angular velocities through the Jacobian matrix. During trajectory planning, quintic polynomial interpolation ensures the continuity and smoothness of the velocity and acceleration curves of each joint. Forward kinematics and Monte Carlo methods are used for simulation analysis, to cover the entire pollination area with the end effector. In the vision system, a binocular camera [15] is used to obtain the 3D coordinates of flowers in real time, combining with the control system for trajectory planning and interpolation point calculation, to precisely control each joint motor and achieve point-to-point precise pollination. However, this robotic arm relies on binocular cameras to obtain the 3D coordinates of the flowers, and strong light or occlusions can affect the accuracy of the vision system.

Traditional algorithms can still achieve certain effects in specific agricultural pollination domains. A recent study by N Duc Tai [16] involved using segmentation methods to locate and segment cantaloupe flowers, employing mathematical models based on the biological characteristics of the cantaloupe flowers to determine the key points of each flower's growth direction, and using an inverse-projection method to convert the position of the flower from a 2D image to a 3D space, thus achieving flower pose localization.

In recent years, with the development and application of deep learning, STRADER J's BrambleBee [4] pollination robot has combined traditional algorithms and deep learning algorithms to estimate flower poses. The system uses a two-stage image-processing framework to recognize flowers and estimate their poses. It first employs a Naive Bayes pixel-level segmentation algorithm, then uses a convolutional neural network for classification and pose refinement, ensuring the accuracy of the recognition results. Finally, an octree-based obstacle map and a factor graph represent the flower map, mapping the flowers and their positions in 3D to optimize the pose estimation. Khubaib Ahmad's automatic watermelon flower pollination robot [17] utilizes deep learning to detect the 2D information of flowers and combines it with traditional algorithms, to calculate the depth of the flower positions, thereby achieving flower localization. The robot then adjusts the mechanical arm, using servo searching, until it reaches the ideal pollination position.

YANG's [18] vision-based servo automated pollination robot uses more advanced target-detection algorithms, such as YOLOv5, YOLACT++ [19], and DETR [20], to detect flowers, identifying the position and orientation of the pistil through rotation object-detection technology. The system then employs a pseudo-binocular ranging strategy, calculating the 3D coordinates of the pistil by moving the camera position and using eye–hand calibration to transform these coordinates into the robot arm's operational coordinate system. During the pollination task, the system combines visual servo control strategies, performing coarse positioning by moving the end effector near the flower, followed by precise positioning using a circular search trajectory to ensure accurate contact between the pollen brush and the pistil. However, the system still has an end effector precision error of 15 mm. Although the circular search trajectory can eventually achieve precise positioning, the process is time-consuming, taking nearly 19 s to complete pollination of a single flower. If the end effector precision can be improved, the efficiency of precise positioning through the circular search trajectory will significantly benefit.

Both the traditional methods of N Duc Tai and the deep learning-based pollination robots of STRADER J, Khubaib Ahmad, and YANG significantly impact the success rate and efficiency of pollination operations through errors in estimating the pose of pollination targets. For instance, the pollination robots of N Duc Tai and YANG can only ensure successful pollination by maintaining a large servo search area, which leads to reduced pollination efficiency. Consequently, our work proposes a deep learning method to learn the translational and rotational errors between the flower and the pollination end effector. By compensating for these errors, the positioning accuracy of the robot's end effector is enhanced. This reduction in the servo mechanism's search range and path ultimately improves the efficiency of the pollination process.

2.3. Transformer Architecture and Position Encoding

The transformer architecture, introduced by Vaswani et al. [21], has significantly impacted Natural Language Processing (NLP) and computer vision. Transformers utilize the self-attention mechanism to capture complex dependencies within input data, making them effective for understanding visual information. In computer vision, transformers process images by segmenting them into patches and linearly embedding these segments, enabling powerful attention mechanisms for tasks like image classification, object detection, and generating image descriptions. In robotics, transformers can predict the error offsets of a robotic arm's end by analyzing sequences of images or sensor data, enhancing precision and accuracy.

The introduction of position encoding [22] in predicting translational offset error and rotational pose offset error at the robotic arm's end further improves spatial awareness and accuracy. By embedding spatial position information directly into the transformer model, it enhances the model's ability to discern the relative and absolute positions of objects in a scene, which is critical for precise error estimation.

The self-attention mechanism of transformers can capture long-range dependencies, allowing the model to directly calculate the relationships between any two elements within an input image feature sequence, considering all the input data rather than just local information. This characteristic helps reduce the issue of local optima. Additionally, through fine-grained attention allocation, transformer-based models can identify and emphasize the features and relationships most relevant to the current task. Therefore, employing a transformer model equipped with positional encoding to predict the translational and rotational errors between the pollination robot's end effector and the target object is a viable approach.

3. Method

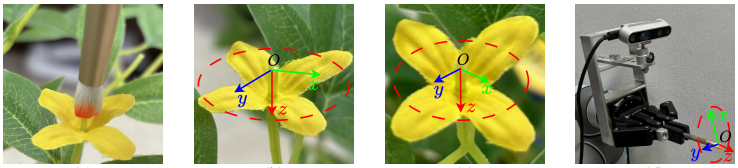
3.1. Offset Error

To describe the translational offset error and the rotational pose offset error, two Cartesian coordinate systems, C_A and C_B, need to be constructed at the target flower and at

the end effector of the pollination robot, respectively. The Cartesian coordinate system C_A, constructed on the flower, has its origin O at the end of the pistil. The plane formed by the x and y axes is parallel to the plane of the flower petals, and the z axis is parallel to the pistil, pointing inward, as shown in Figure 2b,c. The Cartesian coordinate system C_B, constructed at the end of the pollination robot, has its origin O at the end of the brush, obtained by translating the UR5 robotic arm's TCP coordinate system, as shown in Figure 2d. C_A can be derived from C_B through a 4×4 translation–rotation matrix TR:

$$TR = \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_1 \\ R_{21} & R_{22} & R_{23} & t_2 \\ R_{31} & R_{32} & R_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$C_A = TR \cdot C_B \tag{2}$$



(a) (b) (c) (d)

Figure 2. (**a**) The ideal zero-error pollination position. (**b**–**d**) The Cartesian coordinate systems' positions on the flower and at the end of the robotic arm, respectively.

The translational offset error and rotational pose offset error are the translation vector t and the rotational vector derived from the rotation matrix R of the translation–rotation matrix TR, respectively:

$$t = \begin{bmatrix} t_1 & t_2 & t_3 \end{bmatrix}^{\mathrm{T}} \tag{3}$$

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{4}$$

When the end effector of the pollination robot is in the ideal pollination posture, the values of the translational offset error and the rotational pose offset error approach zero. At this moment, the brush at the end of the pollination robot is perpendicular to the plane of the petals of the target flower and just in contact with the pistil, i.e., the coordinate systems C_A and C_B coincide, as shown in Figure 2a.

$$\begin{aligned} \theta &= \cos^{-1}\left(\frac{trace(R) - 1}{2}\right) \\ \overrightarrow{V} &= \frac{1}{2\sin(\theta)} \begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{13} \\ R_{21} - R_{12} \end{bmatrix} \\ \overrightarrow{W} &= \theta \overrightarrow{V} \end{aligned} \tag{5}$$

The translational offset error and rotational pose offset error predicted by the model are denoted as $\hat{t}$ and $\hat{R}$, respectively. The differences between the actual and predicted translational offset error and rotational pose offset error from the end of the robotic arm to the pistil of the target flower are calculated as the translational error (TE) and the rotational error (RE), respectively. According to Equation (5), the rotation matrices R and $\hat{R}$ can be converted into rotational vectors $\overrightarrow{W}$ and $\hat{\overrightarrow{W}}$:

$$TE = \|\hat{t} - t\| \tag{6}$$

$$RE = 1 - \frac{\hat{\overrightarrow{W}} \overrightarrow{W}}{\|\hat{\overrightarrow{W}}\| \|\overrightarrow{W}\|} \tag{7}$$

The unit of TE is centimeters (cm), representing the spatial distance between the predicted translational error and the actual translational error. RE is dimensionless, representing the cosine distance between the predicted rotational vector error and the actual rotational vector error.

3.2. Attention Module

The attention module is implemented based on the self-attention mechanism, which enables the model to weigh the importance of different features within the input data. By applying this mechanism, the model can adaptively concentrate computational resources on regions of the image that contain key information for predicting pose offsets $(\Delta T_x, \Delta T_y, \Delta T_z, \Delta W_x, \Delta W_y, \Delta W_z)$. Furthermore, with the introduction of position encoding in the self-attention mechanism, the model achieves a comprehensive understanding of the image. This provides additional information for addressing symmetry issues in object pose prediction, enhancing the model's accuracy and augmenting its general capability in various scenarios encountered by pollination robots. This method not only improves the accuracy of the model but also strengthens its versatility in the diverse conditions faced by pollination robots.

3.3. Feature Extraction

Given the unique ability of transformer models to process image data, we utilize a pre-trained convolutional neural network, ResNet-50 [23], as the feature extraction network, to convert original images into high-dimensional feature vectors. These features are then fed into the transformer model for further processing. We compare feature extraction networks of different depths, to determine the optimal feature representation. This approach leverages the strong ability of ResNet-50 to capture detailed spatial hierarchies in images, providing a rich set of features for the transformer model to analyze. By incorporating this hybrid architecture, combining the strengths of CNNs in feature extraction with the advanced attention mechanism of transformers, the model achieves a nuanced understanding of image content relevant for predicting the pose offset errors. This method facilitates identification and focusing on crucial aspects of the input data, thereby enhancing the accuracy and efficiency of the pose-estimation process.

3.4. Proposed Approach

This work introduces a novel network model incorporating an attention module, designed to enhance prediction accuracy by focusing on the most critical parts of the input data. This model is particularly suited for analyzing image data captured by the end effector of a pollination robot, aiming to accurately predict the position offset of the end effector relative to the target object in a Cartesian coordinate system. During training, the model takes image data from the robotic arm's end as input, and it outputs a six-dimensional vector $(\Delta T_x, \Delta T_y, \Delta T_z, \Delta W_x, \Delta W_y, \Delta W_z)$ representing the translational offset errors $\Delta T = (\Delta T_x, \Delta T_y, \Delta T_z)$ and the rotational offset errors $\overrightarrow{W} = (\Delta W_x, \Delta W_y, \Delta W_z)$ in the three directions of the Cartesian coordinate system.

3.5. Offset Error Prediction

The input to the model is an RGB image captured by an "eye-in-hand" RGB camera on the robotic arm, depicting the positional state of the robotic arm's end effector during pollination and the pollination flower. The output is a vector representing the pose offset vector $(\Delta T_x, \Delta T_y, \Delta T_z, \Delta W_x, \Delta W_y, \Delta W_z)$, which includes the translational and rotational errors of the robot's end effector relative to the target object. To achieve this, the model employs two feedforward neural networks to directly map the attention-focused features

into the three-dimensional offset space and the three-dimensional rotational space. Figure 3 illustrates the entire algorithmic process.

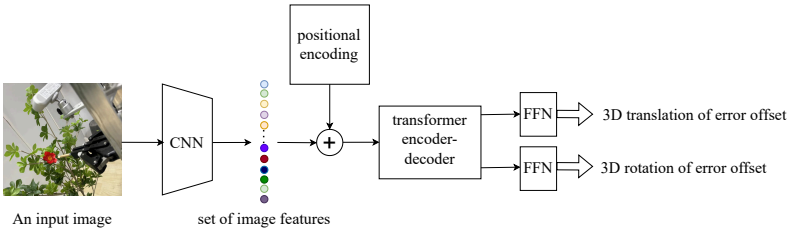


Figure 3. The diagram illustrates the workflow of the proposed algorithm. The process begins with the input image, which undergoes feature extraction via a convolutional neural network (CNN), to obtain high-dimensional image features. These features are then augmented with positional encoding before being input into the transformer module. The transformer's output is further processed by two distinct feedforward neural networks, tasked with predicting translational and rotational errors, respectively.

3.6. Network Architecture

In this work, we have adopted a customized transformer model designed to process serialized image features and predict the translational offset error and rotational pose offset error at the end of a robotic arm. We have modified the original transformer model by introducing two-dimensional positional encoding, to preserve the spatial information of the input images. The output layer is customized, to generate 3D translational offset and a rotational pose offset error vector. As shown in Figure 4, a ResNet50 serves as the backbone for extracting a rich feature set from the input image data. To retain the spatial information among the elements in the images, we apply sinusoidal position encoding to the extracted features, which are then summed with the features before being fed into the encoder. To capture information from different subspaces, a multi-head attention mechanism with five heads is employed in the encoder. The output of the encoder is subsequently fed into the decoder. Following the standard architecture of the transformer, the decoder utilizes a multi-head attention mechanism to transform two embeddings of size d. These query objects are transformed by the decoder into output embeddings, which are then decoded by two separate feedforward networks into the translational offset error and the rotational pose offset error, respectively.

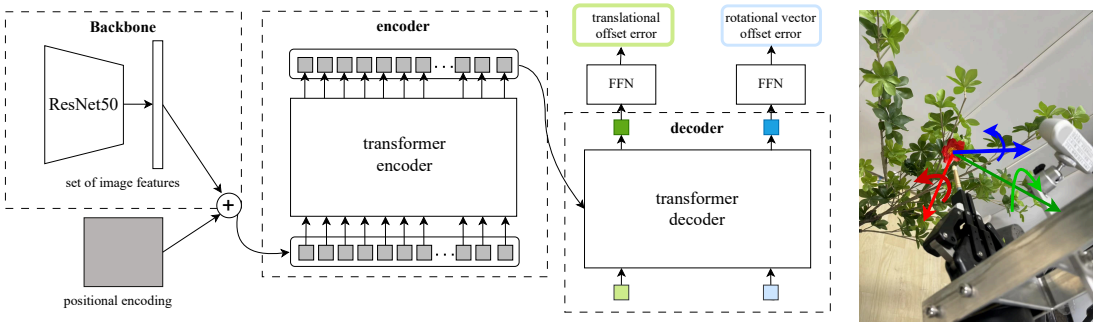


Figure 4. The final model employs ResNet50 as the backbone network to learn high-level features from the input image. These features are then supplemented with positional encoding before being passed to the encoder. Subsequently, the decoder first outputs a feature vector for the translational offset error, which is used as input for a feedforward neural network designed for prediction. This feature vector is then used as the query input for the decoder, resulting in another feature vector that is fed into a separate feedforward neural network for predicting the rotational pose offset error.

4. Experiments

4.1. Data Preprocessing

The image data used in this experiment are all from a proprietary dataset, which consists of images captured by a camera mounted on the robotic arm during the pollination process, where each image uniquely represents the pose of a flower to be pollinated. As shown in Figure 5, the orientation of the flowers relative to the end of the robotic arm was categorized into five directions: left, right, upwards, downwards, and front. Detailed statistical information is available in Table 1.

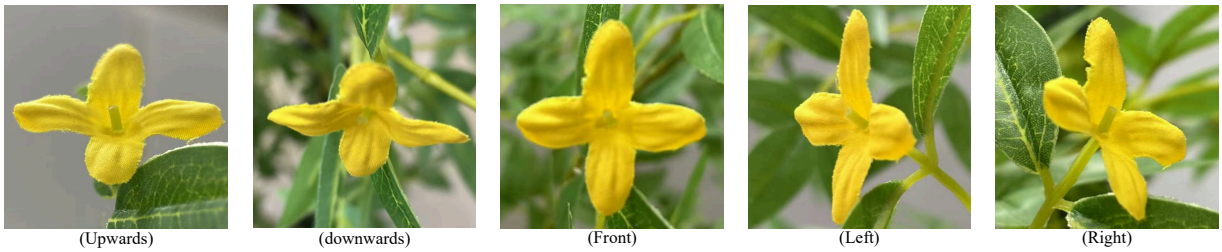


Figure 5. Flowers with different orientations relative to the end effector of the robotic arm.

Table 1. The dataset includes the number of images of flowers oriented in different directions relative to the the end effector of the robotic arm.

Orientations	Number of Flowers	Proportion (%)
L	261	20.6
R	274	21.7
F	230	18.2
U	256	20.3
D	243	19.2

"F", "L", "R", "U", and "D" represent front, left, right, upwards, and downwards orientations.

To ensure the consistency of the input data, all images captured by the robotic arm's end camera were first resized to a uniform resolution of (224×224). Subsequently, the images were normalized, scaling the pixel values to the $[0, 1]$ range, to enhance the stability of the model training. Moreover, a series of data-augmentation techniques, including scaling, cropping, and color transformation, were applied, to increase data diversity and prevent overfitting. Regarding the label data, we assumed the position offset error $\Delta T = (\Delta T_x, \Delta T_y, \Delta T_z)$ and the rotational offset error $\Delta \overrightarrow{W} = (\Delta W_x, \Delta W_y, \Delta W_z)$, with $||\Delta T|| < D$, where D was a constant. The position offset error Equation (8) and rotational offset error Equation (9) were normalized and scaled to $[-1, 1]$, as follows:

$$\Delta T_n = \frac{\Delta T}{D} \tag{8}$$

$$\overrightarrow{W} = \theta \overrightarrow{V} \tag{9}$$

$$\theta = \sqrt{\Delta W_x^2 + \Delta W_y^2 + \Delta W_z^2} \tag{10}$$

$$\overrightarrow{V} = (\frac{\Delta W_x}{\theta}, \frac{\Delta W_y}{\theta}, \frac{\Delta W_z}{\theta}) \tag{11}$$

The symbol θ in the equation (10) represents the rotation angle, and $\overrightarrow{V}$ denotes the unit vector along the axis of rotation. The normalized label data were $(\Delta T_n, \overrightarrow{V}, \frac{\theta}{2\pi})$.

4.2. Evaluation Metrics

In the model presented in this paper, we separately predict the translational offset error and the rotational pose offset error, thus designing two loss functions. The first loss

function, named $Loss_T$, measures the mean squared distance between the spatial position offset error predicted by the model for the end effector of the pollination robotic arm to the pistil of the target flower and the actual spatial offset error. $Loss_T$ is defined as follows:

$$Loss_T = \frac{1}{2m} \sum_{x \in \mathcal{M}} (\hat{T}_n - T_n)^2 \tag{12}$$

where $\mathcal{M}$ is the set of the test dataset, m is the number of elements in the set, and $\hat{T}_n$ and T_n are the translational offset error predicted by the model and the true translational offset error obtained via Equation (8), respectively.

The second loss function, named $Loss_R$, measures the discrepancy between the spatial rotational error predicted by the model for the end effector of the pollination robotic arm to the pistil of the target flower and the actual spatial rotational error. $Loss_R$ is defined as follows:

$$Loss_R = \frac{1}{2m} \sum_{x \in \mathcal{M}} \left(\frac{1}{\sigma_1^2} (\|\hat{\theta} - \theta\|^2) + \frac{1}{\sigma_2^2} \left(1 - \frac{\overrightarrow{\hat{V}} \overrightarrow{V}}{\| \overrightarrow{\hat{V}} \| \| \overrightarrow{V} \|}\right) + \log(\sigma_1 \sigma_2) \right) \tag{13}$$

where $\mathcal{M}$ is the set of the test dataset and m is the number of elements in the set. Variables $\hat{\theta}$, θ, $\overrightarrow{\hat{V}}$, and $\overrightarrow{V}$ represent the predicted and actual rotational angles and axes obtained via Equations (9)–(11), with σ_1 and σ_2 being the parameters that need to be learned.

The combined loss function used for model training is defined as

$$Loss = \alpha Loss_T + \beta Loss_R \tag{14}$$

This loss function in Equation (14) comprehensively measures the translational offset error loss and the rotational offset error loss during model training. Parameters α and β are hyperparameters representing weights that need to be systematically adjusted during model training.

4.3. Training Details

The model training was conducted in a computational environment equipped with NVIDIA V100 GPUs. We used the Adam optimizer [24], with the initial learning rate set to 0.01, and we employed a learning rate decay strategy that gradually reduced the learning rate to 0.00001 as training progressed. A custom loss function, Equation (14), was used during the training process. In the model training, the setting of the hyperparameters in Equation (14) affected the model's ability to converge. After extensive experiments, we finally set $\alpha = 0.0025$ and $\beta = 1$, which allowed the model to converge more easily during training.

In the dataset, flowers with different orientations were randomly shuffled, and then the data were divided into 10 subsets, using the K-fold cross-validation method. Each time, one subset was used as the test set, and the remaining nine subsets were used as the training set. This process was repeated 10 times. During the model training, we observed that the model initially converged rapidly and then gradually stabilized. Figure 6 shows the changes in the values of loss, translational error, and rotational error during the model training process:

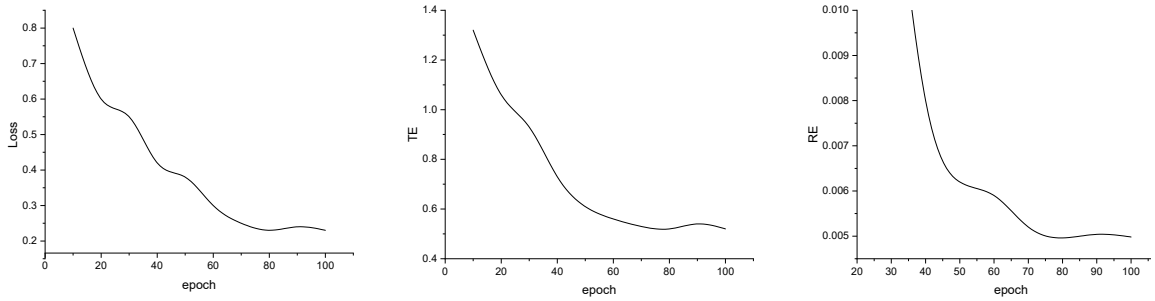


Figure 6. Changes in loss, *TE*, and *RE* during the model training process.

4.4. Results

To the best of our knowledge, we are the first to predict the translational offset error and rotational pose offset error of a robotic arm's end effector relative to a target object using only RGB image information. Therefore, this paper focuses on the accuracy improvements of the translational offset error and rotational pose offset error of the robot's end effector when using our proposed method compared to the state-of-the-art YANG method [18] as the baseline. Additionally, we highlight the enhancement in the efficiency of pollinating a single flower.

We conducted experiments on flowers with different orientations in groups, to analyze translational and rotational offset errors as well as detection speed. The experimental results showed slight variations for flowers with different orientations. As shown in Table 2 and Figure 7, flowers facing forward achieved the best results, in terms of experimental accuracy, with the smallest translational offset error and rotational pose offset error compared to flowers in other orientations. Flowers facing upwards came next. Due to environmental symmetry, there was almost no difference in results for flowers facing left and right. Flowers facing downwards had the worst results, in terms of both translational offset error and rotational pose offset error. However, the detection speed of the model for flowers with different orientations remained almost constant.

Table 2. Experimental results of the model on flowers with different orientations.

Orientations	TE	RE	FPS
L	0.82	0.0049	41
R	0.82	0.0049	42
F	0.78	0.0047	43
U	0.80	0.0048	42
D	0.87	0.0051	41

"F", "L", "R", "U", and "D" represent front, left, right, upwards, and downwards orientations.

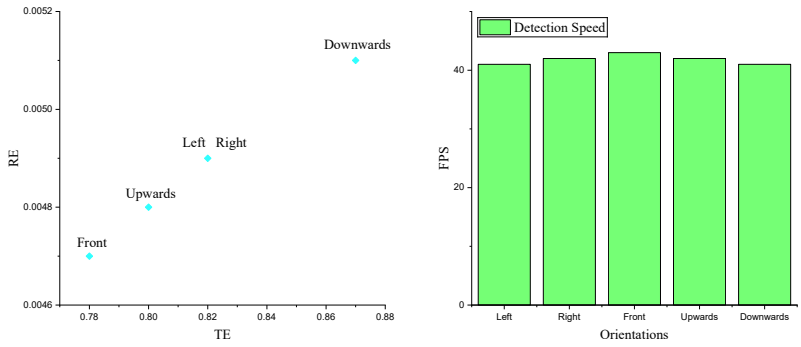


Figure 7. Distribution of the model's rotational and translational error detection for flowers with different orientations, as well as the detection speed.

We conducted our experiments using the YANG pollination robot. The YANG pollination robot's process of pollinating a single flower can be divided into five steps in

chronological order. After the fourth step, the positioning accuracy of the YANG pollination robot's end effector relative to the target pistil was 1.5 cm. At this point, we introduced a new step called "Fine adjustment position". We input the image of the flower's pose relative to the robot's end effector into our trained model, to predict the translational and rotational offset errors. Based on the predicted values, the pose of the pollination robot's end effector was adjusted.

As shown in Table 3, thanks to the new "Fine adjustment position" step, the translational distance and rotational discrepancy between the pollination robot's end effector and the pollination target were further reduced, narrowing the range for the next servo search pollination step. Our experiments showed that the average time for servo search pollination was only 3.1 s, achieving a pollination success rate comparable to YANG's pollination robot at 86.19%. As shown in Table 4, after applying our method to the pollination robot, the average distance accuracy between the robot's end effector and the target pistil reached 0.81 cm, an improvement of 46.67%. The rotational pose offset error calculated according to Equation (14) also reached 0.0049. The total time to complete the pollination of a single flower was reduced by nearly half, with an average efficiency improvement of 50.9%.

Table 3. Comparison of the average time cost for each step of the pollination system after incorporating our method.

Step	YANG (Baseline) Time Cost (S)	Our Time Cost (S)
Flower detection	0.0928	0.0927
Pistil identification	0.025	0.024
Position calculation (robot motion included)	1.8	1.8
Flower reaching (robot motion included)	4.2	4.2
Fine adjustment position	/	0.024
Servoing (robot motion included)	12.7	3.1

The "Fine adjustment position" step is an additional step. With this step included, the time spent in the "Servoing" step is greatly reduced, achieving the same pollination success rate in just 3.1 s. All numbers are recorded in seconds.

Table 4. Comparison of translational offset error and rotational pose offset error when applying our algorithm to flowers oriented in different directions compared to the baseline.

Orientations	Method	TE	RE	Time Cost (s)
L	YANG's	1.53	/	18.78
	Our	0.82	0.0050	9.22
R	YANG's	1.52	/	18.80
	Our	0.82	0.0050	9.23
F	YANG's	1.48	/	18.85
	Our	0.80	0.0048	9.22
U	YANG's	1.53	/	18.79
	Our	0.81	0.0049	9.24
D	YANG's	1.55	/	18.83
	Our	0.83	0.0052	9.26

"F", "L", "R", "U", and "D" represent front, left, right, upwards, and downwards orientations. YANG's method served as the baseline.

This paper conducted several ablation experiments, to verify the impact of different backbones and the addition of positional encoding on the model's prediction of the translational offset error and the rotational pose offset error. ResNet50, ResNet18, ResNet101, VGG16, VGG19, DenseNet-121, and DenseNet-201 were selected as the backbone feature extraction networks. For each backbone, experiments were conducted with and without positional encoding. The experimental results, shown in Table 5 and Figure 8, indicate that different backbones and the addition of positional encoding significantly affect the

model's final prediction accuracy. The version of the model with ResNet50 as the backbone feature extraction network and positional encoding added achieved the best performance in predicting the translational offset error and the rotational pose offset error, reaching 8.1 mm and 0.0049, respectively.

Table 5. The impact of different backbones within the model on the accuracy of predicting the translational offset error and the rotational pose offset error.

Backbones	Positional Encoding	TE	RE
ResNet50	✓	0.81	0.0049
	✗	1.19	0.0051
ResNet18	✓	0.92	0.0053
	✗	1.33	0.0054
ResNet101	✓	0.86	0.0050
	✗	1.21	0.0050
VGG16	✓	0.96	0.0053
	✗	1.32	0.0055
VGG19	✓	0.99	0.0054
	✗	1.33	0.0055
DenseNet-121	✓	0.92	0.0054
	✗	1.23	0.0055
DenseNet-201	✓	1.10	0.0055
	✗	1.38	0.0057

A check mark (✓) indicates that the model included positional encoding, while a cross (✗) indicates that the model did not include positional encoding.

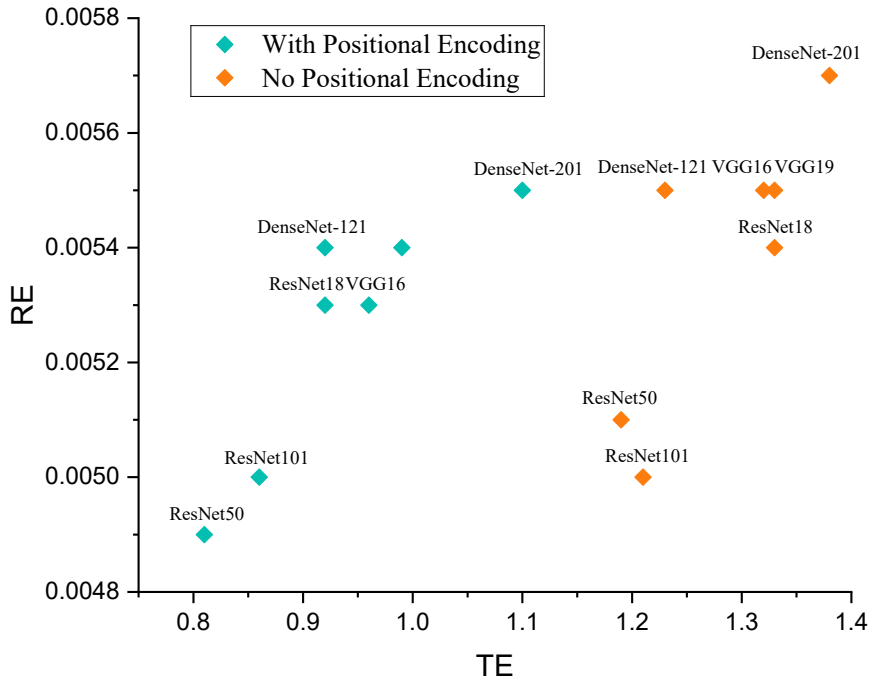


Figure 8. Distribution of translational and rotational error for different model backbones with and without the addition of positional encoding.

5. Discussion

This work proposes a transformer-based approach that achieves end-to-end prediction of translational and rotational errors between the robotic pollination arm's end effector

and the target pollination position, solely using RGB image information. Our experimental results demonstrate that this method effectively narrows the error range of the robotic end effector within a known error margin, thereby enhancing the overall efficiency of the robotic pollination.

In this work, our method exhibited slight variations in results when dealing with flowers of different orientations, particularly larger translational and rotational errors with downwards-facing flowers. This may be attributed to the relative positioning of the robot's camera angle to the flower orientation, complicating the accurate recognition and localization of flowers in specific orientations. Through experimentation, we also observed that the versions of the model that utilized positional encoding performed better in accurately predicting translational and rotational errors. Additionally, the different backbone networks used for extracting features from input images had a significant impact on the model's performance.

This work has potential limitations despite the demonstrated effectiveness of the proposed method in reducing translational and rotational errors between the pollination robot's end effector and the target: (1) The dataset used in this study was specifically collected for experimental purposes, which may limit the model's generalization capabilities across varied environments and flower types; (2) The model's performance may be compromised under different lighting conditions and in obstructed environments, impacting its overall effectiveness. These issues emphasize the need for further enhancements before practical deployment, necessitating future research to explore additional methods that improve the model's adaptability and reliability in diverse agricultural environments.

6. Conclusions

This work presents an innovative method that utilizes the powerful spatial learning and understanding capabilities of a transformer-based deep learning model to achieve end-to-end prediction of translational and rotational errors between the pollination robot's end effector and the target pollination position using only RGB images. Our experimental results demonstrate that this method is effective in further reducing the error range of the pollination robot's end effector within a known error range, thereby improving the overall efficiency of the pollination robot. Future work could focus on investigating the prediction of translational and rotational errors between the robotic end effector and target positions within datasets containing a more diverse range of flower types, under varying lighting conditions and occlusions. Such work would aim to enhance the robustness and generalization capabilities of the model, providing a feasible approach to improving the accuracy of generic robotic end effectors.

Author Contributions: Conceptualization, J.X.; methodology, J.X.; validation, J.X.; formal analysis, J.X., J.C., H.P. and M.Y.; investigation, J.X.; resources, J.X.; data curation, J.X.; writing—original draft preparation, J.X., H.P. and M.Y.; writing—review and editing, J.X., H.P. and M.Y.; visualization, J.X.; supervision, H.P., M.Y. and J.C.; project administration, J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Guangxi Key R&D Plan Project (AB24010164; AB21220038).

Data Availability Statement: The data that support the findings of this work are available from the corresponding author upon reasonable request.

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

Abbreviations

The following abbreviations are used in this manuscript:

TE	translational error
RE	rotational error

References

1. Binns, C. Robotic Insects Could Pollinate Flowers and Find Disaster Victims. *Popular Science*, 17 December 2009.
2. Williams, H.; Nejati, M.; Hussein, S.; Penhall, N.; Lim, J.Y.; Jones, M.H.; Bell, J.; Ahn, H.S.; Bradley, S.; Schaare, P.; et al. Autonomous pollination of individual kiwifruit flowers: Toward a robotic kiwifruit pollinator. *J. Field Robot.* **2020**, *37*, 246–262. [CrossRef]
3. Gao, C.; He, L.; Fang, W.; Wu, Z.; Jiang, H.; Li, R.; Fu, L. A novel pollination robot for kiwifruit flower based on preferential flowers selection and precisely target. *Comput. Electron. Agric.* **2023**, *207*, 107762. [CrossRef]
4. Strader, J.; Nguyen, J.; Tatsch, C.; Du, Y.; Lassak, K.; Buzzo, B.; Watson, R.; Cerbone, H.; Ohi, N.; Yang, C.; et al. Flower interaction subsystem for a precision pollination robot. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 5534–5541.
5. Shaneyfelt, T.; Jamshidi, M.M.; Agaian, S. A vision feedback robotic docking crane system with application to vanilla pollination. I*nt. J. Autom. Control* **2013**, *7*, 62–82. [CrossRef]
6. Yuan, T.; Zhang, S.; Sheng, X.; Wang, D.; Gong, Y.; Li, W. An autonomous pollination robot for hormone treatment of tomato flower in greenhouse. In Proceedings of the 2016 3rd International Conference on Systems and Informatics (ICSAI), Shanghai, China, 19–21 November 2016; pp. 108–113.
7. Abrol, D.P. *Pollination Biology: Biodiversity Conservation and Agricultural Production: Volume 792*; Springer: New York, NY, USA, 2012.
8. Yang, X.; Miyako, E. Soap bubble pollination. *Iscience* **2020**, *23*, 101188. [CrossRef] [PubMed]
9. Hinterstoisser, S.; Cagniart, C.; Ilic, S.; Sturm, P.; Navab, N.; Fua, P.; Lepetit, V. Gradient response maps for real-time detection of textureless objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 876–888. [CrossRef] [PubMed]
10. Cao, Z.; Sheikh, Y.; Banerjee, N.K. Real-time scalable 6dof pose estimation for textureless objects. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 2441–2448.
11. Brachmann, E.; Krull, A.; Michel, F.; Gumhold, S.; Shotton, J.; Rother, C. Learning 6d object pose estimation using 3d object coordinates. In *Proceedings, Part II 13, Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 536–551.
12. Krull, A.; Brachmann, E.; Michel, F.; Yang, M.Y.; Gumhold, S.; Rother, C. Learning analysis-by-synthesis for 6d pose estimation in rgb-d images. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 954–962.
13. Xiang, Y.; Schmidt, T.; Narayanan, V.; Fox, D. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv* **2017**, arXiv:1711.00199.
14. Li, K.; Huo, Y.; Liu, Y.; Shi, Y.; He, Z.; Cui, Y. Design of a lightweight robotic arm for kiwifruit pollination. *Comput. Electron. Agric.* **2022**, *198*, 107114. [CrossRef]
15. Ma, W.P.; Li, W.X.; Cao, P.X. Binocular vision object positioning method for robots based on coarse-fine stereo matching. *Int. J. Autom. Comput.* **2020**, *17*, 562–571. [CrossRef]
16. Tai, N.D.; Trieu, N.M.; Thinh, N.T. Modeling positions and orientations of cantaloupe flowers for automatic pollination. *Agriculture* **2024**, *14*, 746. [CrossRef]
17. Ahmad, K.; Park, J.-E.; Ilyas, T.; Lee, J.-H.; Kim, S.; Kim, H. Accurate and robust pollinations for watermelons using intelligence guided visual servoing. *Comput. Electron. Agric.* **2024**, *219*, 108753. [CrossRef]
18. Yang, M.; Lyu, H.; Zhao, Y.; Sun, Y.; Pan, H.; Sun, Q.; Chen, J.; Qiang, B.; Yang, H. Delivery of pollen to forsythia flower pistils autonomously and precisely using a robot arm. *Comput. Electron. Agric.* **2023**, *214*, 108274. [CrossRef]
19. Zhou, C. *Yolact++ Better Real-Time Instance Segmentation*; University of California: Davis, CA, USA, 2020.
20. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 213–229.
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762v7.
22. Jiang, K.; Peng, P.; Lian, Y.; Xu, W. The encoding method of position embeddings in vision transformer. *J. Vis. Commun. Image Represent.* **2022**, *89*, 103664. [CrossRef]
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
24. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.

electronics

MDPI

Article

Adaptive Aberrance Repressed Correlation Filters with Cooperative Optimization in High-Dimensional Unmanned Aerial Vehicle Task Allocation and Trajectory Planning

Zijie Zheng [1,*], Zhijun Zhang [1], Zhenzhang Li [2], Qiuda Yu [1] and Ya Jiang [1]

[1] School of Automation Science and Engineering, South China University of Technology, Guangzhou 510006, China
[2] School of Mathematics and Systems Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China; zhenzhangli@gpnu.edu.cn
* Correspondence: 201910102802@mail.scut.edu.cn

Abstract: In the rapidly evolving field of unmanned aerial vehicle (UAV) applications, the complexity of task planning and trajectory optimization, particularly in high-dimensional operational environments, is increasingly challenging. This study addresses these challenges by developing the Adaptive Distortion Suppression Correlation Filter Cooperative Optimization (ARCF-ICO) algorithm, designed for high-dimensional UAV task allocation and trajectory planning. The ARCF-ICO algorithm combines advanced correlation filter technologies with multi-objective optimization techniques, enhancing the precision of trajectory planning and efficiency of task allocation. By incorporating weather conditions and other environmental factors, the algorithm ensures robust performance at low altitudes. The ARCF-ICO algorithm improves UAV tracking stability and accuracy by suppressing distortions, facilitating optimal path selection and task execution. Experimental validation using the UAV123@10fps and OTB-100 datasets demonstrates that the ARCF-ICO algorithm outperforms existing methods in Area Under the Curve (AUC) and Precision metrics. Additionally, the algorithm's consideration of battery consumption and endurance further validates its applicability to current UAV technologies. This research advances UAV mission planning and sets new standards for UAV deployment in both civilian and military applications, where adaptability and accuracy are critical.

Keywords: multi-objective optimization; UAV trajectory planning; correlation filters; adaptive algorithms; task allocation

Citation: Zheng, Z.; Zhang, Z.; Li, Z.; Yu, Q.; Jiang, Y. Adaptive Aberrance Repressed Correlation Filters with Cooperative Optimization in High-Dimensional Unmanned Aerial Vehicle Task Allocation and Trajectory Planning. *Electronics* **2024**, *13*, 3071. https://doi.org/10.3390/electronics13153071

Academic Editor: Zhiquan Liu

Received: 29 May 2024
Revised: 20 July 2024
Accepted: 30 July 2024
Published: 2 August 2024

1. Introduction

In the era of the Internet of Things, unmanned aerial vehicles (UAVs) have become essential tools across various domains due to their flexibility and efficiency. As the complexity and diversity of UAV missions increase, the need for advanced high-dimensional multi-objective task allocation and trajectory planning becomes critical. Recent studies by Chen et al. [1–3] have introduced innovative algorithms that enhance the path planning and cooperative behavior control of heterogeneous UAVs. These methods improve the operational efficiency and adaptability of UAVs in dynamic and complex environments, addressing contemporary challenges in UAV deployment.

The rapid development of UAV technology has led to a growing demand for UAV applications in multitasking environments. However, in complex multi-objective environments, a single UAV often struggles to handle multiple tasks simultaneously, necessitating the collaboration of multiple UAVs. Research on high-dimensional multi-objective UAV task allocation and trajectory planning based on deep learning aims to address optimization challenges in multi-UAV collaborative task allocation, achieving intelligent management and scheduling of UAVs to enhance task execution efficiency and accuracy [4].

In the realm of national strategic development, the progression and utilization of unmanned aerial vehicle (UAV) technology emerge as pivotal technological enablers for strategic initiatives. Through the refinement of UAV task planning, pivotal sectors within a nation stand to witness heightened levels of productivity and efficiency, thereby fostering the cultivation of novel, high-quality productivity, a concept championed by Yao and Liu [5]. Their work underscores the importance of delineating theoretical frameworks and practical pathways to drive progress. Moreover, the evolution of UAV technology not only augments a nation's technological capabilities but also fortifies its global competitiveness, as illuminated by Ma and Chen [6]. Their exploration of national transitions underscores the incontrovertible role of technological innovation in shaping economic and strategic trajectories. With UAV technology serving as a cornerstone innovation, its multifaceted applications across domains such as national defense, transportation, and agriculture contribute significantly to enhancing a nation's overall competitiveness. Thus, by aligning UAV technology with broader strategic imperatives, the aspirations of advancing progress and fortifying national competitiveness, as espoused by the new development paradigm, stand poised for realization.

Deep learning models have played a significant role in UAV task allocation and trajectory planning. Common deep learning models include convolutional neural networks (CNNs) [7], recurrent neural networks (RNNs) [8], deep reinforcement learning (DRL) [9], generative adversarial networks (GANs) [10], and transfer learning, each with its advantages and limitations. For instance, CNNs are suitable for image processing tasks but lack efficiency in handling sequential data [11], while RNNs can process sequential data but suffer from issues like vanishing and exploding gradients. DRL can handle tasks with delayed rewards but involves a complex training process, while GANs can generate data but exhibit instability during training. Transfer learning can leverage previously learned knowledge to expedite learning on new tasks but requires addressing domain differences.

This study aims to address high-dimensional multi-objective UAV task allocation and path planning problems using an Adaptive Distortion Suppression Correlation Filter (ARCF). We will develop an ARCF network that integrates the states, actions, and reward functions of UAV missions to enable intelligent decision-making for task allocation and path planning. Specifically, we will employ deep reinforcement learning techniques to train the network, allowing it to learn optimal behavioral strategies for UAVs in complex environments. During the training process, we will utilize replay memory units to store historical experiences and combine target and estimation networks to enhance the stability and efficiency of the learning process.

The main contributions of this paper are as follows:

- Propose a novel Adaptive Distortion Suppression Correlation Filter Cooperative Optimization (ARCF-ICO) algorithm, enhancing the accuracy and stability of UAV mission planning.
- Integrate multi-objective optimization techniques, achieving efficient intelligent decision-making for UAV task allocation and path planning in complex environments.
- Present experimental results which show that the ARCF-ICO algorithm outperforms existing methods in terms of AUC and Precision metrics on UAV123@10fps and OTB-100 datasets.

2. Related Work

2.1. Single-Objective Optimization for Unmanned Aerial Vehicle Trajectory Planning

The trajectory planning for unmanned aerial vehicles (UAVs) is fundamentally an optimization problem with practical implications. Li and Duan [12] incorporated threat cost and fuel cost into a weighted optimization objective and employed an improved universal gravitational search algorithm to enhance the global search convergence, thereby improving the quality of optimal solutions for UAV trajectories. Qu et al. [13] combined a simplified grey wolf optimizer with an enhanced symbiotic organism search to propose a novel hybrid algorithm for obtaining feasible and effective routes. Dasdemir et al. [14] designed a generic

preference-based single-objective evolutionary algorithm to optimize both the total distance of planned routes and radar detection threats. Yao et al. [15] introduced a hybrid algorithm based on a model predictive control and an improved grey wolf optimizer for planning optimal trajectories for UAV target tracking in urban environments. Papaioannou et al. [16] addressed the challenges of passively monitoring multiple moving targets in obstructed environments with UAVs by designing a model predictive guidance controller combined with a joint estimation and control strategy. Ren et al. [17] proposed a multi-objective path planning (MOPP) approach using the Non-dominated Sorting Genetic Algorithm II (NSGA-II), optimized for both distance and safety, demonstrating its efficiency in an urban setting by employing octree-based spatial subdivisions and safety index maps.

However, current research on trajectory planning for both single and multiple UAVs often focuses on single-objective optimization, either considering a single objective or integrating multiple optimization objectives into a single one through linear weighting. Such optimization approaches heavily rely on subjective weighting coefficients set by decision-makers, directly impacting the optimization results, and may overlook trajectories with outstanding performance in relatively minor objectives. In recent years, despite the increasing attention to multi-objective optimization-based trajectory planning problems, typically considering only two to three optimized objectives, the practical optimization requirements for UAV trajectory planning extend beyond a limited number of objectives. Addressing this issue, establishing a trajectory planning model based on high-dimensional multi-objective optimization becomes particularly crucial to simultaneously optimize various performance aspects of trajectories.

2.2. High-Dimensional Multi-Objective Optimization for UAV Trajectory Planning

High-dimensional multi-objective optimization problems are prevalent in both life and engineering practices, where multiple objectives need simultaneous optimization, often with inter-objective correlations leading to conflicting situations. In such cases, alternative optimization schemes need to be considered to ensure the generation of equivalent solutions in the absence of correlated information from other schemes.

Storn and Price [18] proposed the Differential Evolution (DE) algorithm, a population-based method similar to steady-state replacement mechanisms, for solving real-parameter optimization problems. New offspring only compete with their corresponding parents, and if offspring exhibit better fitness, they replace them. With the emergence of new bio-inspired heuristics such as Particle Swarm Optimization [19], Grey Wolf Algorithm [20], Whale Algorithm [21], and Sparrow Search Algorithm [22], understanding how they apply to different types of objective optimization problems becomes crucial. The literature has optimized UAV trajectory distance and trajectory threat cost using genetic algorithms and smoothed them [23].

The demarcation of high-dimensional multi-objective optimization algorithms lies in whether the number of optimized objectives exceeds four [24]. With an increasing number of optimized objectives, the number of non-dominated solutions generated during the solving process of multi-objective optimization algorithms exponentially increases, significantly affecting the algorithm's performance and efficiency [25,26]. Additionally, the selection pressure generated by multi-objective optimization algorithms in solving high-dimensional multi-objective problems is often insufficient to guide individuals in the population towards ideal points.

Strategies to enhance these two indicators mainly fall into three categories: (1) Enhancing the selection pressure of algorithms by changing Pareto dominance methods to accelerate the convergence rate of populations. GrEA [27] uses grid-based evaluation metrics to enhance the selection pressure of algorithms. NSGA-III [28] uses a reference point strategy instead of NSGA-II's crowding distance strategy [29] to select excellent individuals from non-dominated solutions, thereby improving the convergence of algorithms. 1by1EA [30] selects offspring individuals one by one based on individual convergence when environmental selection occurs, then enhances the diversity of populations through niche techniques. RPEA [31] continually

generates a series of well-converged and distributed reference points based on the current population to guide evolution. (2) Decomposing a complex high-dimensional multi-objective optimization problem into a group of sub-problems and co-optimizing these sub-problems. MOEA [32] decomposes the problem into a series of adaptively decomposed single-objective optimization sub-problems and then aggregates information from neighboring problems. MaOEA/Ds [33] uses a set of uniformly distributed self-guided reference vectors in space to divide the decision space into multiple small subspaces and judge the merits of individuals in subspaces. Yi et al. [34] proposed a multi-objective evolutionary optimization method based on objective decomposition, decomposing the problem into several sub-problems and solving each sub-problem in parallel, fully utilizing the information of other sub-populations to improve the selection pressure of non-dominated solutions. (3) Strategy based on evaluation metrics. Evaluating the superiority and inferiority of individuals comprehensively through evaluation metrics, and then selecting excellent individuals for genetic operations. However, the computational complexity of such strategies is usually high, and such evaluation metrics are commonly used to assess the quality of algorithm optimization results [35].

In the process of collaborative trajectory planning for multiple UAVs, it is necessary to consider not only the individual trajectory attributes but also the spatial coordination among multiple UAVs. To avoid the impact of combining multiple optimization objectives into a single one through weighting on trajectory planning, a model based on high-dimensional multi-objective optimization is proposed for collaborative trajectory planning for multiple UAVs. This model optimizes UAV trajectory distance cost, trajectory safety cost, trajectory energy cost, and spatial coordination among multiple UAVs as optimization objectives. Unlike existing approaches that treat the trajectory of a single UAV as an individual in the population and optimize the trajectories of multiple UAVs separately, in this model, multiple UAV trajectories are treated as a whole entity in the population, and optimization is performed simultaneously on multiple UAV trajectories. Additionally, comprehensive evaluations of the convergence and diversity of individuals in the population are conducted, and mating strategies of the algorithm for trajectory planning problems are improved to enhance convergence performance [35,36]. Through the algorithm, a set of Pareto-optimal trajectories for multiple UAVs are obtained for decision-makers to use. Decision-makers can select the most suitable trajectories for their mission attributes from this set of Pareto-optimal trajectories.

3. Multi-UAV Flight Path Planning Modeling

3.1. Problem Description

In the context of collaborative trajectory planning for multiple unmanned aerial vehicles (UAVs), a group of UAVs are tasked to navigate from multiple starting points to a series of specified target points to execute complex missions. The mission scenario is set within an area protected by various defense systems, where UAVs must cleverly evade threats in enemy radar coverage and anti-aircraft fire zones, while considering the performance limitations and cooperation requirements of each UAV. Based on this scenario, we conduct simulation analyses of UAV group navigation and mission execution paths in three-dimensional space. The computational assumptions are as follows:

(1) All UAVs maintain consistent flight speed during mission execution. (2) Each trajectory segment is divided into straight flight paths. (3) Each UAV possesses identical performance characteristics.

The core of collaborative trajectory planning for multiple UAVs involves the design of trajectory cost models and collaborative spatial models. Trajectory cost primarily considers distance, threat, and energy consumption. The objective of the model is to minimize these costs while enhancing spatial coordination among UAVs.

Assuming there are N_z target points, each requiring reconnaissance, strike, and confirmation tasks represented as S_{m_j}, where $j \in [1, N_z]$ denotes the jth target point's mth task type, corresponding to reconnaissance, strike, and confirmation ($m = 1, 2, 3$). Hence, the total number of tasks is $\sum_{n=1}^{N_z} \sum_{m=1}^{3} S_{m_n}$. Each task has specific time windows $[c_{m_n}, d_{m_n}]$, and designated ammunition requirements z_j for strike tasks at each target point.

The UAV group consists of N_p reconnaissance UAVs, N_q strike UAVs, and N_pq-reconnaissance-and-strike-integrated UAVs, totaling $N_v = N_p + N_q + N_pq$ UAVs, indexed as $u \in [1, N_v]$. Each UAV u has a capability vector $Capability_u$ corresponding to its task type, where $Capability_u(m)$ represents the capability of UAV u to perform task type m, taking values of 1 or 0. The payload capacity of each UAV u is z_u, and if UAV u^* lacks strike capability, then $z_{u^*} = 0$. It is assumed that the relationship between fuel consumption fp_u and flight speed V_u of each UAV u is $fp_u = \alpha_u \times V_u$, where α represents the proportionality between fuel consumption and flight speed, and V_u is within the range $[k_u, j_u]$, denoting the minimum and maximum speeds.

In urban environments, the problem of multiple UAVs tracking multiple ground targets is depicted in Figure 1, where N UAVs are ready to execute M tasks. The UAV set is denoted as $U = \{U_1, U_2, \ldots, U_N\}$, and the target set is denoted as $H = \{H_1, H_2, \ldots, H_M\}$, with task types represented by $M_i = \{1, 2\}$ ($M_i = 1$ for reconnaissance, $M_i = 2$ for strike).

To demonstrate UAV heterogeneity and the specific performance requirements of mission scenarios, Z_{u_m} is used to represent the UAV performance matrix, where matrix elements represent the UAV's ability to execute a certain task. For example, if there are three UAVs and their performance matrix Z_{u_m} is shown in Table 1, $Z_{u_m}(1,1) = 0.9$ indicates that UAV 1's reconnaissance task capability is 0.9. If H_1 is a reconnaissance task ($M_1 = 1$) with a minimum requirement capability of 0.6, then among the three UAVs U_1, U_2, U_3, only U_1 (0.9) can meet the task requirements of H_1.

Table 1. UAV capability value.

Unmanned Aerial Vehicle	Ability Value	
	Scout	Track
U_1	0.9	0.4
U_2	0.3	0.9
U_3	0.5	0.5

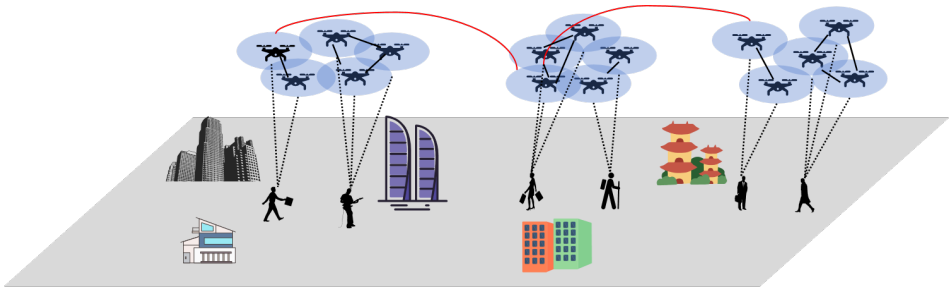


Figure 1. Multi-UAV task allocation in urban scenarios.

3.2. Optimization Design Objective Model

Our goal is to minimize the total fuel consumption of the entire UAV fleet during task execution while maximizing the efficiency of task completion. Therefore, we define the following cost functions.

Total Fuel Consumption Cost: This represents the total fuel consumption of UAVs from takeoff, task execution, to return to the airport. The fuel consumption of each UAV depends on the flight distance and speed. If we define the fuel consumption of UAV u performing task k from point P_i to P_j as $f_{u,k,i,j}$, then the total fuel consumption F_{fuel} can be expressed as

$$F_{fuel} = \sum_{u=1}^{N_u} \sum_{k=1}^{3} \sum_{i=0}^{N_p} \sum_{j=1}^{N_p} X_{u,k,i,j} \cdot f_{u,k,i,j} \tag{1}$$

Task Time Fitness Cost: This measures the difference between the completion time of tasks and the midpoint of their time windows. The goal is to make the UAVs' task

execution times as close to the midpoint of the task time windows as possible. If we denote the time window fitness of UAV u as $T_{adapt,u}$, then the total time window fitness F_{time} is

$$F_{time} = \sum_{u=1}^{N_u} T_{adapt,u} = \sum_{u=1}^{N_u} \left| \sum_{i=0}^{N_p} \sum_{j=1}^{N_p} \sum_{k=1}^{3} X_{u,k,i,j} \cdot (t_{u,k,i,j} - \frac{s_{k,j} + e_{k,j}}{2}) \right| \tag{2}$$

Combining the above two cost functions, we form a bi-objective optimization problem:

$$\text{Minimize} \quad F = \alpha \cdot F_{fuel} + \beta \cdot F_{time} \tag{3}$$

where α and β are parameters that balance the importance of the two objectives. In Equation (3), the objective functions are normalized to ensure that each contributes equally to the overall optimization. The normalization process involves scaling each objective function to a $[0, 1]$ range based on their respective maximum and minimum values observed during the initial runs.

3.3. Constraints

In the UAV trajectory planning problem, we must consider the following constraints, including weather conditions to account for low-altitude operations.

Task Execution Constraint: Ensure that each task is executed by at least one UAV with the corresponding capability. The constraint formula for task execution is as follows:

$$\sum_{u=1}^{N_u} X_{u,k,i,j} \geq 1 \quad \forall k, \forall i, \forall j \tag{4}$$

Here, $X_{u,k,i,j}$ represents the binary decision variable indicating whether UAV u executes task k from point P_i to P_j. N_u is the total number of UAVs. k represents the task type. i represents the starting point index. j represents the ending point index.

Capability Matching Constraint: The tasks performed by UAVs must comply with their capability limitations. The constraint formula for capability matching is as follows:

$$\sum_{u=1}^{N_u} X_{u,k,i,j} \geq 1 \quad \forall k, \forall i, \forall j X_{u,k,i,j} \leq Capability_{u,k} \quad \forall u, \forall k, \forall i, \forall j \tag{5}$$

where $Capability_{u,k}$ represents the capability of UAV u to perform task k.

Weather Conditions Constraint: Operations at low altitudes are influenced by weather conditions such as wind speed, precipitation, and visibility. These factors are incorporated into the trajectory estimation to ensure safe and reliable operations. The constraint formula for weather conditions is as follows:

$$W_{u,k,i,j} \leq W_{max} \quad \forall u, \forall k, \forall i, \forall j \tag{6}$$

where $W_{u,k,i,j}$ represents the weather impact on UAV u while performing task k from point P_i to P_j. W_{max} is the maximum allowable weather impact.

Flight Time Constraint: Ensure that the time for a UAV to fly from one task point to the next does not exceed the specified maximum value. The constraint formula for flight time is as follows:

$$t_{u,k,i,j} - t_{u,k,i,j-1} \leq T_{max} \quad \forall u, \forall k, \forall i, \forall j > 1 \tag{7}$$

where $t_{u,k,i,j}$ represents the time at which UAV u arrives at task point j while performing task k, starting from point P_i. T_{max} is the maximum allowable flight time between consecutive task points.

3.4. Performance Metrics

Performance evaluation is crucial to validate the effectiveness of our optimization model. We utilize the following performance metrics:

Task Coverage: The ratio of successfully assigned and completed tasks to the total number of tasks. Task coverage can be expressed as

$$\text{Coverage} = \frac{\text{Number of successfully assigned and completed tasks}}{\text{Total number of tasks}} \tag{8}$$

Average Battery Consumption: The average battery consumption of UAV formation to perform all tasks is expressed as follows:

$$\text{Average Battery Consumption} = \frac{\sum_{u=1}^{N_u} \text{Battery used by } u}{N_u} \tag{9}$$

Endurance: Endurance measures the operational time of the UAVs, ensuring they can complete tasks within their battery limits. It is expressed as

$$\text{Endurance} = \frac{\sum_{u=1}^{N_u} \text{Operational time of } u}{N_u} \tag{10}$$

Time Efficiency: The difference between the completion time of all tasks and the earliest start time. It can be expressed as

$$\text{Time Efficiency} = \max_{u \in U}\{t_{\text{completion},u}\} - \min_{u \in U}\{t_{\text{start},u}\} \tag{11}$$

Through these performance metrics, we can comprehensively evaluate the effectiveness of the optimization algorithm, enabling further adjustments to model parameters or algorithmic improvements.

4. Adaptive Genetic Multi-Objective Estimation Algorithm

The ARCF (Adaptive Response Map Correction Filter) algorithm aims to integrate the response map distortion occurring during the tracking process with the training process of the filter, thereby enhancing the algorithm's performance (as depicted in Figure 2). To suppress response map distortion, the first step is distortion identification (i.e., determine when response map distortion occurs). It introduces the Euclidean norm to define the difference between the previous frame M_1 and the current frame M_2 response maps.

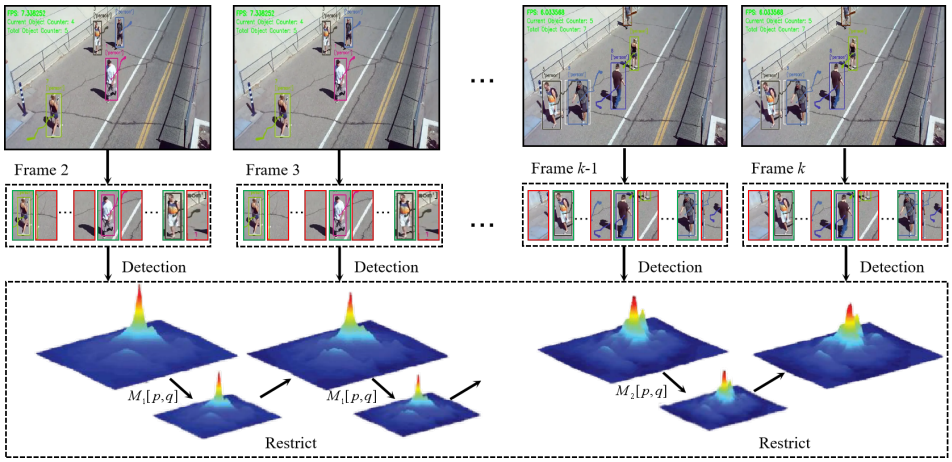


Figure 2. Flowchart of the ARCF algorithm.

The primary objective of the ARCF algorithm is to combine the response map distortion generated during target tracking with the filter training process, enhancing tracking performance by dynamically updating the filter. Response map distortion is primarily caused by the target's rapid motion or external environmental factors such as occlusion and illumination changes. The ARCF algorithm identifies and suppresses these distortions by analyzing the changes in response maps between consecutive frames, specifically by computing the Euclidean distance between the response maps of two frames.

In the algorithm, the response maps of two consecutive frames F_1 and F_2 are denoted as M_1 and M_2, respectively, and aligned using the shift operation ψ to calculate their differences. The formula for calculating the difference is

$$\Delta M = \psi(M_1, M_2) = \sqrt{\sum_{p,q}(M_1[p,q] - M_2[p,q])^2} \tag{12}$$

Here, p and q represent the spatial coordinates of the response map.

Based on the aforementioned difference measure, the objective function of the ARCF algorithm can be described as an optimization problem, aiming to minimize response map distortion while maximizing the accuracy of target tracking. The objective function consists of a distortion term and a regularization term, expressed as

$$\min_h \left\{ \lambda \|h\|^2 + \gamma \sum_{d=1}^{D} \|h * M_d - Y_d\|^2 + \epsilon \Delta M \right\} \tag{13}$$

where h is the filter, $*$ denotes the convolution operation, M_d is the input response map for the d-th channel, Y_d is the desired output response map, and λ, γ, and ϵ are coefficients adjusting the importance of each term.

For computational efficiency, the objective function is further transformed into the frequency domain. In the frequency domain, the formula becomes

$$\min_H \left\{ \lambda \|H\|^2 + \gamma \|\mathcal{F}(H) \odot X - Y\|^2 + \epsilon \Delta M \right\} \tag{14}$$

where $\mathcal{F}$ represents the Fourier transform, $\odot$ denotes element-wise multiplication, and X and Y are the input and desired output frequency–domain representations, respectively.

By employing the ADMM (Alternating Direction Method of Multipliers) algorithm to solve the optimization problem, the filter can be effectively updated. The solution process involves two main sub-problems: optimizing the filter and updating the response map. This approach effectively suppresses response map distortion caused by rapid motion or external environmental changes, thereby improving the stability and accuracy of the tracking algorithm. The ARCF algorithm can be summarized as Algorithm 1.

Algorithm 1: The Procedure of ARCF

Initialize filter h_0, learning rates $\lambda, \gamma, \epsilon$
Set maximum iterations T
for $t = 1$ to T **do**
  Capture current frame F_t
  Compute response map M_t using h_{t-1}
  if $t > 1$ **then**
    Compute distortion $\Delta M = \sqrt{\sum_{p,q}(M_t[p,q] - M_{t-1}[p,q])^2}$
  end if
  Update filter in frequency domain:
  $H_t = \arg\min_H \lambda \|H\|^2 + \gamma \|\mathcal{F}(H) \odot X_t - Y_t\|^2 + \epsilon \Delta M$
  Update response map M_t:
  $M_t = H_t * F_t$
  Check convergence:
  if $\|\mathcal{F}(H_t) - \mathcal{F}(H_{t-1})\| <$ threshold **then**
    break
  end if
  Update h_t with H_t in spatial domain
end for
Output the global best solution $G_{\text{best}} = h_T$

4.1. ARCF-ICO Algorithm

The ARCF-ICO algorithm integrates significant enhancements to the original ARCF strategy, focusing on both solution convergence and diversity within a high-dimensional optimization space. This is particularly pertinent in the context of UAV mission planning, where diverse operational environments and rapidly evolving mission requirements necessitate a robust and adaptive optimization approach.

In the initial phase of the ARCF-ICO algorithm, priority is given to achieving high convergence rates. This ensures a rapid alignment of UAVs towards optimal trajectories or solution sets, effectively addressing immediate operational needs such as surveillance or threat detection. As the algorithm progresses, emphasis shifts towards preserving diversity among solutions. This is crucial in UAV operations to explore a range of potential flight paths or tactical strategies, thereby avoiding local optima and enhancing mission outcome robustness. The ARCF-ICO algorithm is designed to be implemented by various categories of UAVs, including fixed-wing, rotary-wing, and hybrid UAVs. These UAVs can be used in both commercial and military applications, depending on their capabilities and mission requirements.

The comprehensive evaluation indicator for convergence and diversity (CAD) [37] is a newly proposed metric designed to assess both the convergence and diversity of solutions within our ARCF-ICO framework. This novel indicator is defined as follows:

$$\mathrm{CAD}(u_i, U) = \left[1 + \mathrm{rand}(0.8, 1) \times M \times \left(\frac{t}{t_{\max}}\right)^{\theta} \times D(u_i, U)\right] \times (1 - C(u_i, U)) \tag{15}$$

where $D(u_i, U)$ represents the diversity measure and $C(u_i, U)$ denotes the convergence measure for UAV u_i within the fleet U. The parameter M denotes the number of objectives, and θ governs the balance between convergence and diversity as the algorithm iterates from t to $t_{\max}$—the maximum number of generations.

Diversity $D(u_i, U)$ is computed as

$$\mathrm{SDE}(u_i, U) = \min_{u_i \in U, j \neq i} \sqrt{\sum_{k=1}^{m} \mathrm{sde}(f'_k(u_i, U), f'_k(u_j, U))^2} \tag{16}$$

where $\mathrm{sde}(f'_k(u_i, U), f'_k(u_j, U))$ is defined as:

$$\mathrm{sde}(f'_k(u_i, U), f'_k(u_j, U)) = \begin{cases} f'_k(u_j, U) - f'_k(u_i, U) & f'_k(u_j, U) > f'_k(u_i, U) \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

Convergence $C(u_i, U)$ of a UAV in relation to the fleet is quantified as

$$C(u_i, U) = \frac{\mathrm{Disc}(u_i, U)}{\sqrt{m}} \tag{18}$$

where $\mathrm{Disc}(u_i, U)$ represents the Euclidean distance from UAV u_i to the ideal solution in the normalized objective space.

By integrating these strategies, ARCF-ICO dynamically adapts UAV mission planning and response tactics according to evolving environmental conditions and operational demands, optimizing both the efficiency and effectiveness of deployed UAVs.

4.2. ARCF-ICO Multi-Objective Planning

In the ARCF-ICO multi-objective optimization model, we have devised an efficient mating strategy tailored for the complex task environments encountered by UAVs. This strategy combines the favorable traits of individuals within the population and introduces stochastic elements to increase population diversity, thereby enhancing the adaptability and flexibility of the algorithm. The ARCF-ICO algorithm is applicable to missions conducted in Visual Line Of Sight (VLOS) [38], Beyond Visual Line Of Sight (BVLOS) [38], and fully

autonomous operations. The flexibility of the algorithm allows it to adapt to different operational constraints and requirements.

Within the context of UAV trajectory planning, the trajectory points generated by each UAV for each trajectory segment are represented by the following matrix:

$$X = [X_1, X_2, \ldots, X_m]^T \tag{19}$$

where X_i is an $N \times 3L$ matrix representing the ith individual in the population, N is the number of UAVs, and L is the number of trajectory segments.

During the mating process in the parent population, two parents P_1 and P_2 are selected based on their comprehensive evaluation indicator CAD. The CAD calculation formula is as follows:

$$\mathrm{CAD}(P) = \frac{1}{N}\sum_{i=1}^{N}\left[1 + \mathrm{rand}(0.8, 1) \times M \times \left(\frac{t}{t_{\max}}\right)^{\theta} \times D(P_i, P)\right] \times (1 - C(P_i, P)) \tag{20}$$

where $D(P_i, P)$ and $C(P_i, P)$ represent the diversity and convergence indicators of individual P_i, M is the number of objective functions, θ is the balance parameter, t is the current generation, and $t_{\max}$ is the maximum generation.

The mating operation is as follows, combining the trajectory information of the two parents to generate new offspring trajectories:

$$P_{\mathrm{new},j} = \alpha P_{1,j} + (1 - \alpha) P_{2,j} \tag{21}$$

where $P_{1,j}$ and $P_{2,j}$ denote the position coordinates of parents P_1 and P_2 at the jth trajectory point, respectively, and α is a random number between 0 and 1 used to control the contribution ratio of the two parents in the newly generated offspring. The individual mating operation is shown in Figure 3.

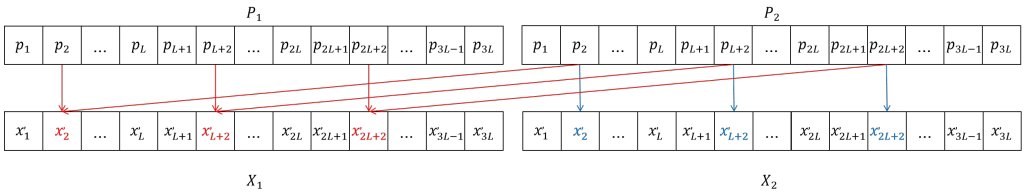


Figure 3. Individual mating operation.

This design ensures that UAVs can adjust their flight strategies according to actual task requirements and environmental changes when executing tasks such as surveillance, reconnaissance, or other complex missions. Furthermore, the multi-objective optimization approach allows UAVs to optimize other important task metrics, such as flight time and fuel efficiency, while ensuring task efficiency, thereby ensuring comprehensive and high-performance mission execution.

5. Simulation and Result Analysis

5.1. Datasets

To comprehensively evaluate the performance of the algorithm, we utilize two widely used datasets: the UAV123@10fps dataset [39] and the OTB-100 dataset [40]. Below are the specific details of each dataset:

UAV123@10fps Dataset: This dataset comprises 123 tracking scenarios captured using UAVs in aerial environments. It includes a mix of real-world and synthetic scenes generated using simulators. The dataset covers 12 different tracking challenge environments, providing a diverse set of scenarios for evaluation.

OTB-100 Dataset: The OTB-100 dataset consists of 100 real-world tracking scenarios captured manually. It encompasses 11 distinct tracking challenge environments, offering a wide range of scenarios to assess algorithm performance.

The utilization of these datasets enables a comprehensive evaluation of the algorithm across various tracking challenges and environmental conditions. Its specific information is shown in Table 2 below.

Table 2. UAV123@10fps and the OTB-100 dataset details.

UAV123@10fps			OTB-100	
Serial Number	**Challenge Name**	**Number**	**Challenge Name**	**Number**
1	Scale Variation (SV)	109	Scale Variation (SV)	65
2	Aspect Ratio Change (ARC)	68	Occlusion (OCC)	49
3	Background Clutter (BC)	21	Illumination Variation (IV)	38
4	Camera Motion (CM)	70	Motion Blur (MB)	31
5	Fast Motion (FM)	28	Deformation (DEF)	43
6	Full Occlusion (FOC)	33	Fast Motion (FM)	43
7	Illumination Variation (IV)	31	Out-of-Plane Rotation (OPR)	64
8	Low Resolution (LR)	48	In-Plane Rotation (IPR)	52
9	Out-of-View (OV)	30	Background Clutters (BCs)	33
10	Partial Occlusion (POC)	73	Out-of-View (OV)	14
11	Similar Object (SOB)	39	Low Resolution (LR)	10
12	Viewpoint Change (VC)	60	-	-

5.2. Experimental Parameter Setting

In the setup of the experiments, the hardware configuration used includes an Intel Core i9-13700 processor and 32 GB of memory. The software configuration is based on the MATLAB R2019a platform. Regarding parameter settings, the regularization parameter is set at 1.2, following the settings from the original ARCF algorithm. After experimental adjustments, another regularization parameter is set to 0.001. The learning rate for the target template, denoted by η, is set to 0.0192, again following the guidelines from the original ARCF algorithm.

5.3. Correlation Evaluation Indices

For the ARCF-ICO multi-objective optimization algorithm in UAV mission planning, we employ three improved evaluation metrics to comprehensively assess algorithm performance: Area Under the Curve (AUC), Center Location Error (CLE), and Precision. These metrics effectively gauge the performance and accuracy of UAVs during mission execution.

Area Under the Curve (AUC): This metric evaluates the overall success rate of UAVs across multiple flight tasks, particularly in maintaining target objectives in complex environments. AUC is calculated as

$$\mathrm{AUC} = \frac{\mathrm{area}(B_{\mathrm{pred}} \cap B_{\mathrm{true}})}{\mathrm{area}(B_{\mathrm{pred}} \cup B_{\mathrm{true}})}, \tag{22}$$

where B_{pred} represents the predicted target location rectangle by the UAV algorithm, and B_{true} represents the rectangle of the true target location.

Center Location Error (CLE): This metric measures the average Euclidean distance between the predicted center position of the UAV and the true center position of the target. High accuracy in CLE is essential for ensuring precise execution of tasks such as monitoring and reconnaissance. CLE is calculated as

$$\mathrm{CLE} = \sqrt{(x_{\mathrm{pred}} - x_{\mathrm{true}})^2 + (y_{\mathrm{pred}} - y_{\mathrm{true}})^2}, \tag{23}$$

where $(x_{\mathrm{pred}}, y_{\mathrm{pred}})$ is the predicted center position by the UAV, and $(x_{\mathrm{true}}, y_{\mathrm{true}})$ is the true center position of the target.

Precision: Precision measures the accuracy of UAV localization within a specific threshold t, i.e., the proportion of frames where the prediction error of the target's center position is within the threshold. This is a critical metric for evaluating the real-time tracking performance of UAVs. Precision is calculated as

$$\text{Precision} = \frac{N_{t \leq \text{threshold}}}{N_{\text{total}}}, \tag{24}$$

where $N_{t \leq \text{threshold}}$ is the number of frames where CLE is less than or equal to the threshold t, and N_{total} is the total number of frames. The threshold t is typically set to 20 pixels.

Through these three evaluation metrics, the performance of UAVs in executing tasks in complex environments, such as path planning accuracy and stability, can be comprehensively assessed. These metrics not only aid in optimizing UAV operational strategies but also provide crucial feedback for further algorithm improvement and flight parameter adjustments.

5.4. Comparison Study

In this chapter, the ARCF-ICO algorithm is evaluated using two datasets: UAV123@10fps and OTB-100. To better understand the performance of the proposed ARCF-ICO algorithm, it will be compared with eight popular algorithms in the field of video object tracking, including KCF [41], LDES [42], MCCT-H [43], Staple [44], fDSST [45], and AutoTrack [46]. LDES and AutoTrack algorithms were published in AAAI2019 and CVPR2020, respectively, focusing on UAV datasets, while the remaining five algorithms are classic correlation filter-based tracking algorithms in recent years.

Figure 4 illustrates the comprehensive comparison of the ARCF-ICO algorithm with the other six mainstream algorithms in the UAV123@10fps dataset in terms of AUC and Precision. From the figure, it can be observed that the proposed ARCF-ICO algorithm achieves the highest performance, with a comprehensive AUC of 0.516 and comprehensive Precision of 0.712, outperforming the other six algorithms. The AutoTrack and LDES algorithms rank second and third, with comprehensive AUC values of 0.504 and 0.492, and comprehensive Precision values of 0.682 and 0.655, respectively. Compared to the ARCF-ICO algorithm, AutoTrack and ARCF algorithms have lower AUC by 1.2% and 2.4%, and lower Precision by 3.0% and 5.7%, respectively. The remaining five algorithms, MCCT-H, Staple, fDSST, and KCF, have lower AUC values, ranging from 0.285 to 0.456, and lower Precision values, ranging from 0.384 to 0.581, compared to the ARCF-ICO algorithm.

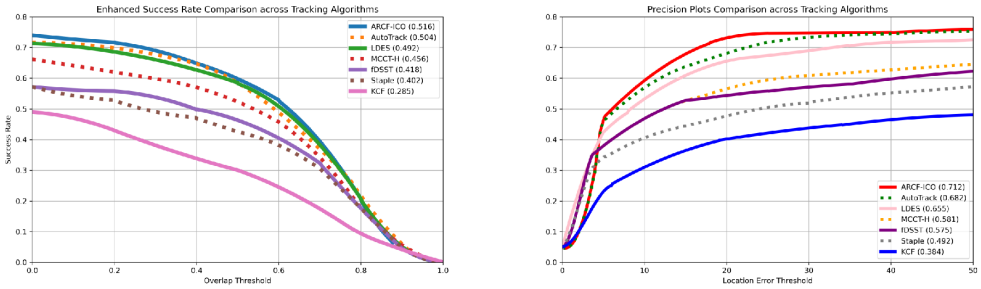


Figure 4. Comparison between AUC and Precision in the UAV123@10fps dataset.

Table 3 presents the comprehensive Precision comparison of various algorithms in the 12 challenging environments of the UAV123@10fps dataset. It can be observed from Table 3 that the ARCF-ICO algorithm consistently ranks first in Precision across all 12 challenging environments. The AutoTrack and LDES algorithms rank second in ten and two challenging environments, respectively. Thus, the comparison results from Table 3 demonstrate that the ARCF-ICO algorithm adapts well to most complex tracking challenges and exhibits robust tracking performance.

Table 3. UAV123@10fps Precision comparison table for various challenge scenarios of the dataset.

Challenge Name	Our Algorithm	Contrast Algorithm					
	ARCF-ICO	AutoTrack	LDES	MCCT-H	fDSST	Staple	KCF
SV	0.724	0.672	0.642	0.545	0.521	0.496	0.372
ARC	0.692	0.686	0.631	0.591	0.552	0.503	0.336
BC	0.702	0.621	0.603	0.503	0.521	0.467	0.415
CM	0.684	0.691	0.633	0.512	0.488	0.472	0.375
FM	0.696	0.664	0.628	0.498	0.508	0.475	0.311
FOC	0.693	0.677	0.689	0.582	0.571	0.436	0.388
IV	0.669	0.673	0.642	0.603	0.598	0.479	0.406
LR	0.688	0.645	0.667	0.654	0.635	0.527	0.416
OV	0.741	0.714	0.656	0.672	0.626	0.426	0.403
POC	0.744	0.725	0.702	0.625	0.645	0.539	0.369
SOB	0.752	0.714	0.693	0.564	0.586	0.545	0.388
VC	0.763	0.702	0.671	0.625	0.645	0.535	0.426

Table 4 shows the comparative results of various algorithms' comprehensive Precision across 11 challenging environments within the OTB-100 dataset. From Table 4, it is apparent that the ARCF-ICO algorithm ranks first in Precision in all 11 environments, with the AutoTrack and LDES algorithms ranking second in five and three environments, respectively. Additionally, the MCCT-H and Staple algorithms achieve second place in the Fast Motion and Out-of-Plane Rotation challenges, respectively. Hence, the comparison results from Table 4 further validate the effectiveness of the ARCF-ICO algorithm.

Table 4. OTB-100 Precision comparison table for various challenge scenarios of the dataset.

Challenge Name	Our Algorithm	Contrast Algorithm					
	ARCF-ICO	AutoTrack	LDES	MCCT-H	fDSST	Staple	KCF
FM	0.517	0.485	0.497	0.505	0.422	0.388	0.263
BC	0.483	0.476	0.496	0.476	0.412	0.433	0.314
MB	0.514	0.502	0.467	0.443	0.402	0.367	0.278
DEF	0.482	0.477	0.472	0.425	0.378	0.336	0.266
IV	0.541	0.538	0.533	0.501	0.445	0.388	0.325
IPR	0.477	0.432	0.375	0.468	0.305	0.325	0.213
LR	0.533	0.532	0.596	0.492	0.462	0.439	0.306
OCC	0.488	0.482	0.457	0.426	0.430	0.422	0.239
OPR	0.563	0.552	0.512	0.423	0.458	0.563	0.325
OV	0.534	0.529	0.505	0.412	0.433	0.368	0.336
SV	0.545	0.537	0.546	0.447	0.448	0.392	0.268

5.5. Simulation Visualization

Figure 5 illustrates the task allocation results obtained by the ARCF-ICO algorithm, demonstrating that this dynamic allocation problem is essentially a nonlinear programming problem with optimal solutions. The left panel of Figure 5 shows the Pareto solutions for UAV task allocation and trajectory planning derived from the UAV123@10fps dataset, while the right panel presents the solutions from the OTB-100 dataset. The Pareto front in both panels indicates the trade-offs between different objectives, showcasing the efficiency and effectiveness of the ARCF-ICO algorithm in handling multi-objective optimization. In addition, we also show the MATLAB simulation task assignment results of 10 UAVs in

Figure 6. As can be seen from Figure 6d, 10 UAVs are about to find the corresponding target object.

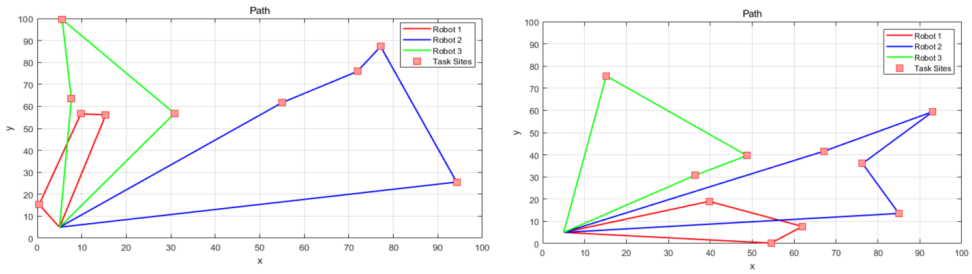


Figure 5. ARCF-ICO algorithm solution task assignment. Pareto solutions for UAV task allocation and trajectory planning derived from the UAV123@10fps dataset (**left**) and the OTB-100 dataset (**right**).

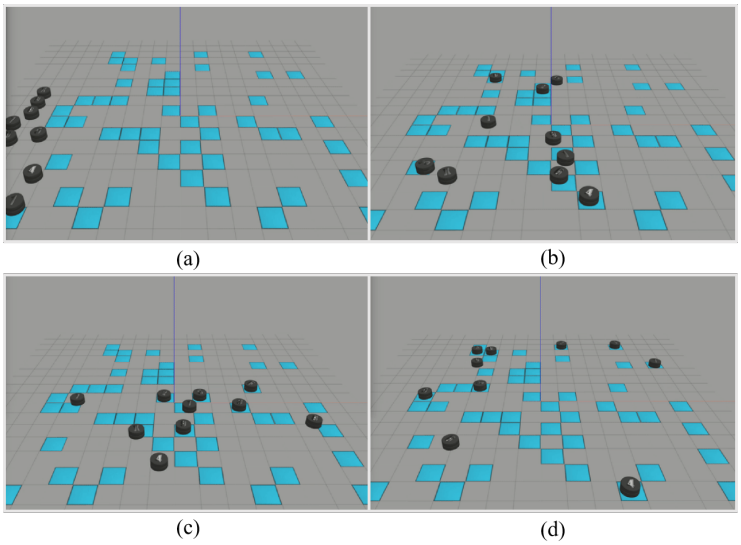


Figure 6. MATLAB simulation task assignment demonstration of 10 UAVs. (**a**) The initial state of the UAV; (**b**,**c**) Intermediate status of UAV task assignment; (**d**) Final status of UAVs assignment.

Furthermore, simulation tests are conducted on scenes from the OTB-100 dataset. The environment map is divided into five sub-maps, but using the ARCF-ICO algorithm alone for UAV path planning may not achieve the fastest model training speed. This could be due to the continuous trial-and-error learning process of the ARCF original algorithm in the environment, which requires more time. Figure 7 illustrates the trajectory planning of UAV tasks in the cropped environment maps of the OTB-100 dataset using the ARCF-ICO algorithm.

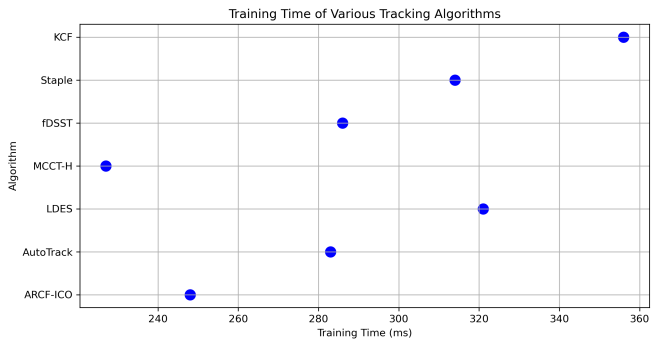


Figure 7. The ARCF-ICO algorithm is demonstrated in the cropping environment map of OTB-100 dataset for UAV mission planning. Blue dot represents the training time for each algorithm.

6. Conclusions and Discussion

This study presents the ARCF-ICO algorithm to address multi-objective optimization in UAV mission planning, enhancing UAV tracking performance in complex environments. By focusing on both convergence and diversity, the ARCF-ICO algorithm adapts to rapid environmental changes and dynamic mission demands through real-time filter updates. Validation using the UAV123@10fps and OTB-100 datasets demonstrates that the ARCF-ICO algorithm outperforms existing methods in AUC and Precision metrics, indicating superior tracking accuracy and robustness. However, two primary limitations are identified: the algorithm's suboptimal performance with extremely high-speed dynamic targets and the need for improved computational efficiency. Future research will explore more efficient algorithmic structures to reduce computational resource consumption while optimizing the response to high-speed moving targets. Overall, the ARCF-ICO algorithm offers a significant technological advancement for multi-objective optimization in UAVs, providing substantial theoretical and practical value for both civilian and military applications.

Author Contributions: Conceptualization, methodology, writing—original draft preparation, visualization, and supervision: Z.Z. (Zijie Zheng). Validation, formal analysis, and investigation: Z.Z. (Zhijun Zhang). Software, validation, writing—review and editing: Z.L. Resources, data curation, writing—review and editing: Q.Y. Validation, formal analysis, and methodology: Y.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: UAV123 dataset download address: https://cemse.kaust.edu.sa/ivul/uav123 (accessed on 10 March 2024). The OTB dataset download address: http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html (accessed on 10 March 2024). The code supporting the findings of this study is not publicly available but can be obtained from the corresponding author upon reasonable request.

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

References

1. Chen, J.; Zhang, Y.; Wu, L.; You, T.; Ning, X. An adaptive clustering-based algorithm for automatic path planning of heterogeneous UAVs. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 16842–16853. [CrossRef]
2. Chen, J.; Ling, F.; Zhang, Y.; You, T.; Liu, Y.; Du, X. Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system. *Swarm Evol. Comput.* **2022**, *69*, 101005. [CrossRef]
3. Chen, J.; Li, T.; Zhang, Y.; You, T.; Lu, Y.; Tiwari, P.; Kumar, N. Global-and-Local Attention-Based Reinforcement Learning for Cooperative Behaviour Control of Multiple UAVs. *IEEE Trans. Veh. Technol.* **2024**, *73*, 4194–4206. [CrossRef]
4. Ning, X.; Tian, W.; He, F.; Bai, X.; Sun, L.; Li, W. Hyper-sausage coverage function neuron model and learning algorithm for image classification. *Pattern Recognit.* **2023**, *136*, 109216. [CrossRef]
5. Yao, Y.; Liu, Z. The New Development Concept Helps Accelerate the Formation of New Quality Productivity: Theoretical Logic and Implementation Paths. *J. Xi'an Univ. Financ. Econ.* **2024**, *37*, 3–14.
6. Ma, G.; Chen, X. From Financial Power to Financial Powerhouse: International Comparison and China's Approach. *J. Xi'an Univ. Financ. Econ.* **2024**, *37*, 46–59.
7. Wang, J.; Li, F.; An, Y.; Zhang, X.; Sun, H. Towards Robust LiDAR-Camera Fusion in BEV Space via Mutual Deformable Attention and Temporal Aggregation. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 5753–5764. [CrossRef]
8. Chen, C.C.; Wei, C.C.; Chen, S.H.; Sun, L.M.; Lin, H.H. AI predicted competency model to maximize job performance. *Cybern. Syst.* **2022**, *53*, 298–317. [CrossRef]
9. Kaufmann, E.; Bauersfeld, L.; Loquercio, A.; Müller, M.; Koltun, V.; Scaramuzza, D. Champion-level drone racing using deep reinforcement learning. *Nature* **2023**, *620*, 982–987. [CrossRef]
10. Uthamacumaran, A. Pattern Detection on Glioblastoma's Waddington Landscape via Generative Adversarial Networks. *Cybern. Syst.* **2022**, *53*, 223–237. [CrossRef]
11. Ning, E.; Wang, C.; Zhang, H.; Ning, X.; Tiwari, P. Occluded person re-identification with deep learning: A survey and perspectives. *Expert Syst. Appl.* **2024**, *239*, 122419. [CrossRef]
12. Li, P.; Duan, H. Path planning of unmanned aerial vehicle based on improved gravitational search algorithm. *Sci. China Technol. Sci.* **2012**, *55*, 2712–2719. [CrossRef]
13. Qu, C.; Gai, W.; Zhang, J.; Zhong, M. A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning. *Knowl.-Based Syst.* **2020**, *194*, 105530. [CrossRef]

14. Dasdemir, E.; Köksalan, M.; Öztürk, D.T. A flexible reference point-based multi-objective evolutionary algorithm: An application to the UAV route planning problem. *Comput. Oper. Res.* **2020**, *114*, 104811. [CrossRef]
15. Yao, P.; Wang, H.; Ji, H. Multi-UAVs tracking target in urban environment by model predictive control and Improved Grey Wolf Optimizer. *Aerosp. Sci. Technol.* **2016**, *55*, 131–143. [CrossRef]
16. Papaioannou, S.; Laoudias, C.; Kolios, P.; Theocharides, T.; Panayiotou, C.G. Joint Estimation and Control for Multi-Target Passive Monitoring with an Autonomous UAV Agent. In Proceedings of the 2023 31st Mediterranean Conference on Control and Automation (MED), Limassol, Cyprus, 26–29 June 2023; pp. 176–181.
17. Ren, Q.; Yao, Y.; Yang, G.; Zhou, X. Multi-objective path planning for UAV in the urban environment based on CDNSGA-II. In Proceedings of the 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE), San Francisco, CA, USA, 4–9 April 2019; pp. 350–3505.
18. Price, K.V. An introduction to differential evolution. In *New Ideas in Optimization*; McGraw: Berkshire, UK, 1999; pp. 79–108.
19. Gad, A.G. Particle swarm optimization algorithm and its applications: A systematic review. *Arch. Comput. Methods Eng.* **2022**, *29*, 2531–2561. [CrossRef]
20. Yu, X.; Jiang, N.; Wang, X.; Li, M. A hybrid algorithm based on grey wolf optimizer and differential evolution for UAV path planning. *Expert Syst. Appl.* **2023**, *215*, 119327. [CrossRef]
21. Liu, J.; Wei, X.; Huang, H. An improved grey wolf optimization algorithm and its application in path planning. *IEEE Access* **2021**, *9*, 121944–121956. [CrossRef]
22. Xue, J.; Shen, B.; Pan, A. An intensified sparrow search algorithm for solving optimization problems. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 9173–9189. [CrossRef]
23. Sahingoz, O.K. Flyable path planning for a multi-UAV system with Genetic Algorithms and Bezier curves. In Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 28–31 May 2013; pp. 41–48.
24. Cui, Z.; Zhang, J.; Wu, D.; Cai, X.; Wang, H.; Zhang, W.; Chen, J. Hybrid many-objective particle swarm optimization algorithm for green coal production problem. *Inf. Sci.* **2020**, *518*, 256–271. [CrossRef]
25. Zhang, J.; Xue, F.; Cai, X.; Cui, Z.; Chang, Y.; Zhang, W.; Li, W. Privacy protection based on many-objective optimization algorithm. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e5342. [CrossRef]
26. Gong, D.; Sun, J.; Miao, Z. A set-based genetic algorithm for interval many-objective optimization problems. *IEEE Trans. Evol. Comput.* **2016**, *22*, 47–60. [CrossRef]
27. Yang, S.; Li, M.; Liu, X.; Zheng, J. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **2013**, *17*, 721–736. [CrossRef]
28. Cui, Z.; Chang, Y.; Zhang, J.; Cai, X.; Zhang, W. Improved NSGA-III with selection-and-elimination operator. *Swarm Evol. Comput.* **2019**, *49*, 23–33. [CrossRef]
29. Hobbie, J.G.; Gandomi, A.H.; Rahimi, I. A comparison of constraint handling techniques on NSGA-II. *Arch. Comput. Methods Eng.* **2021**, *28*, 3475–3490. [CrossRef]
30. Liu, Y.; Gong, D.; Sun, J.; Jin, Y. A many-objective evolutionary algorithm using a one-by-one selection strategy. *IEEE Trans. Cybern.* **2017**, *47*, 2689–2702. [CrossRef] [PubMed]
31. Gao, X.; Zhang, H.; Song, S. A regularity property-driven evolutionary algorithm for multiobjective optimization. *Swarm Evol. Comput.* **2023**, *78*, 101258. [CrossRef]
32. Bao, C.; Gao, D.; Gu, W.; Xu, L.; Goodman, E.D. A new adaptive decomposition-based evolutionary algorithm for multi-and many-objective optimization. *Expert Syst. Appl.* **2023**, *213*, 119080. [CrossRef]
33. Liu, S.; Lin, Q.; Wong, K.C.; Coello, C.A.C.; Li, J.; Ming, Z.; Zhang, J. A self-guided reference vector strategy for many-objective optimization. *IEEE Trans. Cybern.* **2020**, *52*, 1164–1178. [CrossRef]
34. Yi, J.; Zhang, W.; Bai, J.; Zhou, W.; Yao, L. Multifactorial evolutionary algorithm based on improved dynamical decomposition for many-objective optimization problems. *IEEE Trans. Evol. Comput.* **2021**, *26*, 334–348. [CrossRef]
35. Nondy, J.; Gogoi, T.K. Performance comparison of multi-objective evolutionary algorithms for exergetic and exergoenvironomic optimization of a benchmark combined heat and power system. *Energy* **2021**, *233*, 121135. [CrossRef]
36. Cai, X.; Hu, Z.; Chen, J. A many-objective optimization recommendation algorithm based on knowledge mining. *Inf. Sci.* **2020**, *537*, 148–161. [CrossRef]
37. Cai, T.; Wang, H. A general convergence analysis method for evolutionary multi-objective optimization algorithm. *Inf. Sci.* **2024**, *663*, 120267. [CrossRef]
38. Davies, L.; Bolam, R.C.; Vagapov, Y.; Anuchin, A. Review of unmanned aircraft system technologies to enable beyond visual line of sight (BVLOS) operations. In Proceedings of the 2018 X International Conference on Electrical Power Drive Systems (ICEPDS), Novocherkassk, Russia, 3–6 October 2018; pp. 1–6.
39. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for uav tracking. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 445–461.
40. Wu, Y.; Lim, J.; Yang, M.H. Online object tracking: A benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2411–2418.
41. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef]

42. Li, Y.; Zhu, J.; Hoi, S.C.; Song, W.; Wang, Z.; Liu, H. Robust estimation of similarity transformation for visual object tracking. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8666–8673.
43. Wang, N.; Zhou, W.; Tian, Q.; Hong, R.; Wang, M.; Li, H. Multi-cue correlation filters for robust visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4844–4853.
44. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H. Staple: Complementary learners for real-time tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1401–1409.
45. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Discriminative scale space tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1561–1575. [CrossRef]
46. Li, Y.; Fu, C.; Ding, F.; Huang, Z.; Lu, G. AutoTrack: Towards high-performance visual tracking for UAV with automatic spatio-temporal regularization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2020; pp. 11923–11932.

electronics

MDPI

Article

Optimizing Sensor Placement and Machine Learning Techniques for Accurate Hand Gesture Classification

Lakshya Chaplot [1], Sara Houshmand [1], Karla Beltran Martinez [1], John Andersen [2,3] and Hossein Rouhani [1,3,*]

[1] Department of Mechanical Engineering, University of Alberta, Edmonton, AB T6G 2R3, Canada; lakshyachaplot@gmail.com (L.C.); shoushma@ualberta.ca (S.H.); beltranm@ualberta.ca (K.B.M.)
[2] Department of Pediatrics, University of Alberta, Edmonton, AB T6G 2R3, Canada; john.andersen@albertahealthservices.ca
[3] Glenrose Rehabilitation Hospital, Edmonton, AB T5G 0B7, Canada
* Correspondence: hrouhani@ualberta.ca

Abstract: Millions of individuals are living with upper extremity amputations, making them potential beneficiaries of hand and arm prostheses. While myoelectric prostheses have evolved to meet amputees' needs, challenges remain related to their control. This research leverages surface electromyography sensors and machine learning techniques to classify five fundamental hand gestures. By utilizing features extracted from electromyography data, we employed a nonlinear, multiple-kernel learning-based support vector machine classifier for gesture recognition. Our dataset encompassed eight young nondisabled participants. Additionally, our study conducted a comparative analysis of five distinct sensor placement configurations. These configurations capture electromyography data associated with index finger and thumb movements, as well as index finger and ring finger movements. We also compared four different classifiers to determine the most capable one to classify hand gestures. The dual-sensor setup strategically placed to capture thumb and index finger movements was the most effective—this dual-sensor setup achieved 90% accuracy for classifying all five gestures using the support vector machine classifier. Furthermore, the application of multiple-kernel learning within the support vector machine classifier showcases its efficacy, achieving the highest classification accuracy amongst all classifiers. This study showcased the potential of surface electromyography sensors and machine learning in enhancing the control and functionality of myoelectric prostheses for individuals with upper extremity amputations.

Keywords: myoelectric sensor; hand gesture; support vector machine; prosthetic hand; classification; machine learning

Citation: Chaplot, L.; Houshmand, S.; Martinez, K.B.; Andersen, J.; Rouhani, H. Optimizing Sensor Placement and Machine Learning Techniques for Accurate Hand Gesture Classification. *Electronics* **2024**, *13*, 3072. https://doi.org/10.3390/electronics13153072

Academic Editor: Janos Botzheim

Received: 7 June 2024
Revised: 24 July 2024
Accepted: 1 August 2024
Published: 3 August 2024

1. Introduction

In 2017, the global count of unilateral upper limb amputees exceeded 11.3 million, with an additional 11.0 million individuals experiencing bilateral upper limb amputations [1]. Within Canada, around 6800 individuals live with an amputation proximal to the wrist [2]. A recent study [2] compared the utility outcomes and costs associated with two interventions for treating hand amputations: hand vascularized composite allotransplantation and myoelectric hand prostheses. The conclusion was that treating unilateral amputations with myoelectric prostheses was more cost-effective.

Myoelectric prosthetic hands have emerged as a pivotal avenue for restoring both gesture and prehensile capabilities in upper-limb amputees, offering a non-invasive alternative to permanent surgical interventions [2]. Prevalent control systems in prostheses often utilize a trigger-based mechanism relying on one or two surface electromyography (EMG) channels [3,4]. This setup maps single muscle contraction events to predefined movement sequences, necessitating explicit user commands for mode switching [3,4]. This sequential switching introduces latency in execution times, requiring multiple distinct commands to transition between different grip modes [3,4]. The non-intuitive nature of

this grip-switching process, coupled with awkward control dynamics and a lack of sufficient feedback, has been identified as a primary contributor to the low acceptance rates observed for myoelectric prosthesis devices [5,6]. Addressing these challenges is essential for enhancing the usability and acceptance of myoelectric prosthetic devices within the user community [5,6].

Several endeavors to classify sEMG (surface electromyography) signals from human forearm muscles have been documented in previous works. To mitigate intuitiveness concerns, prototype solutions within the existing literature focus on deciphering user gesture intentions by targeting distinct flexor and extensor muscles in the forearm [7]. The voluntary contractions of the remaining forearm muscles after amputation can be identified through machine learning classifiers, such as artificial neural network (ANN), linear discriminant analysis (LDA), and support vector machine (SVM) classifiers. SVM is often chosen due to its mathematical interpretability and global optimization. It performs well even with a small training set [8]. SVM's elasticity parameter, also known as the box constraint hyperparameter *C*, controls the maximum penalty imposed on margin-violating observations and aids in preventing overfitting [8]. Palkowski and Redlarski [9] employed two EMG sensors on the forearm, sampling data at 16 Hz, to discern six whole hand and wrist gestures using an SVM classifier. Lee et al. [10] successfully classified ten hand gestures using features obtained by three EMG sensors and achieved an accuracy exceeding 90% for each participant. However, their machine learning model underwent training and testing on participant datasets without cross-participant data amalgamation to assess the model's generalization capability. This restrictive testing approach introduces bias into the classification accuracy, raising questions about the applicability of this technology in prosthetic hands. Other previous works [8,11,12] achieved high accuracy of over 90% for classifying multiple whole hand and wrist gestures, like wrist opening/closing, ulnar and radial deviation, and flexion-extension. However, whole hand and wrist gestures are less challenging to classify and offer a limited functional application for upper limb amputees seeking the restoration of manual dexterity.

The efficacy of targeting specific muscles requires the strategic deployment of electrodes and their configuration—a critical consideration for feature-based approaches that are relatively unexplored in the literature.

This project aimed to improve hand gesture classification accuracy, strategic sensor placement, and the selection of practical gestures for potential integration with myoelectric prosthetic hands. Therefore, this study investigated the development of a multiple kernel learning (MKL)-based SVM classifier for classifying five intricate hand gestures crucial for amputees—power grasp (clenching of the wrist), hook grip (four-digit grasp), fine pinch (using index finger and thumb), coarse pinch (using all five digits), and point gesture (flexing of digit 3, 4, and 5). The research methodology involved building the classifier using data collected across five distinct sensor configurations, each utilizing one or two EMG sensors on the forearm toward a minimalistic data collection approach, while striving to identify the optimal sensor placement to obtain the highest classification accuracy.

2. Materials and Methods

2.1. Experimental Procedure

sEMG data were collected from the right hand of eight participants without neurological/musculoskeletal impairment or diagnosis (age: 21 $\pm$ 2 years (mean $\pm$ SD), body height: 169.5 $\pm$ 2.8 cm (mean $\pm$ SD), body weight: 57.9 $\pm$ 9.5 kg (mean $\pm$ SD), 7 males and 1 female). All participants were acquainted with the experimental procedures. Informed consent was obtained from all subjects involved in the study. The study was conducted in accordance with the Declaration of Helsinki and approved by the Institutional Ethics Committee of the University of Alberta (AB T6G 2N2, approved on 25 April 2022).

The sEMG signals were acquired using bipolar MyoWare 2.0 Muscle sensors [13] (SparkFun Electronics, Niwot, CO, USA) chosen for their potential for integration into low-cost hand prostheses. The previous version of this sensor has been frequently used in

the literature because of its low cost, easy-to-customize features, and favorable performance reports in validation studies, showing it to be comparable to more expensive commercial EMG systems [14,15]. MyoWare 2.0 has three electrodes—mid-muscle, end-muscle, and reference [13]. Before being acquired by the microcontroller, the differential signal passed through an instrumentation amplifier with a high CMRR (common mode rejection ratio, which is the ratio of differential gain to common-mode gain of the amplifier stage) (140 dB) and unitary gain to eliminate common noise sources, such as the 50 Hz line noise. Subsequently, the signal was filtered by a first-order band-pass filter with cut-off frequencies at 20 Hz and 498 Hz [13]. Following this, the signal underwent rectification and smoothing, achieved by a low-pass envelope detection circuit (3.6 Hz) embedded in the sensor hardware, resulting in a linear envelope of the EMG signal. The enveloped EMG output was then acquired by the microcontroller (Arduino UNO) with a sampling frequency of 780 Hz (SD = 5Hz) to address concerns regarding undersampling in previous studies [16].

The EMG sensor placement configurations were based on the gestures aimed to be classified. The sensors were placed only along those forearm muscles majorly responsible for flexion of the index finger, ring finger, and thumb. The five gestures used were (1) a coarse pinch using all five digits, (2) a fine pinch using index finger and thumb, (3) a hook grip (four-digit grasp), (4) point gesture (flexing of digits 3, 4, and 5), and (5) a power grasp (clenching of the wrist), labelled as 0 to 4, respectively, for the classifier. These five gestures (Figure 1) are considered to have high utility for amputees in their daily lives and are also seen in commercial prosthetics [3,4].

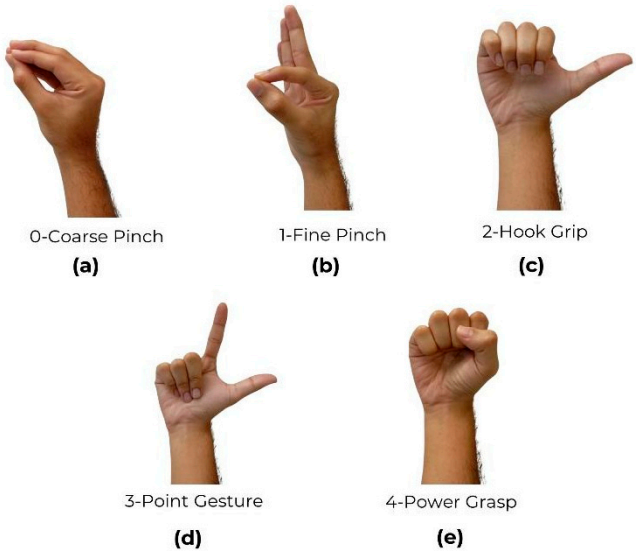


Figure 1. The five different gestures with their corresponding prediction labels: (**a**) coarse pinch, (**b**) fine pinch, (**c**) hook grip, (**d**) point gesture, and (**e**) power grasp.

The EMG sensors were placed at three locations on the ventral side of the forearm (Figure 2) for five different configurations to acquire sEMG signals from muscles primarily responsible for the flexion of the index finger, ring finger, and thumb. The sensor configurations were based on [17–19] and are described as follows:

- C_1: One sensor placed proximal to the wrist along the flexor pollicis longus to acquire the thumb flexion data.
- C_2: One sensor placed proximal to the wrist along flexor digitorum superficialis to acquire the index finger flexion data.
- C_3: One sensor placed along flexor digitorum superficialis and flexor digitorum profundus to acquire ring finger flexion data.
- C_4: Two sensors placed proximal to the wrist along flexor digitorum superficialis to acquire index finger flexion data and proximal to the elbow along flexor digitorum superficialis and flexor digitorum profundus to acquire ring finger flexion data.

- C_5: Two sensors placed proximal to the wrist along the flexor digitorum superficialis to acquire index finger flexion data and along the flexor pollicis longus to acquire the thumb flexion data.

The participants were asked to perform each gesture repeatedly 20 times, with a time interval of two seconds between each repetition. After performing one round of data collection for a single gesture, the participant took a 4 min rest to relax their muscles to prevent fatigue before beginning the next round for a new gesture. The signals measured from the two EMG sensors were recorded simultaneously.

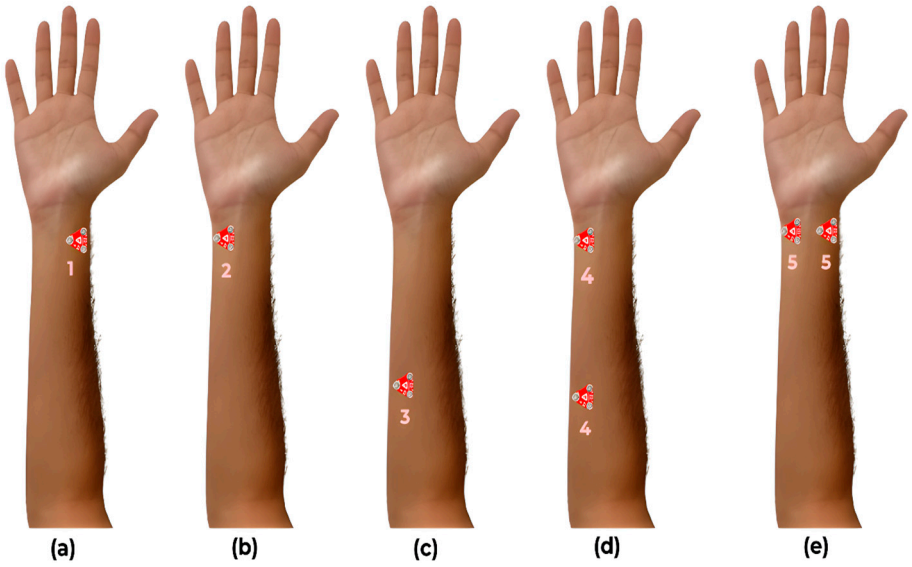


Figure 2. The five different sensor placement configurations used for acquiring EMG data: (**a**) thumb flexion data, (**b**) index finger flexion data, (**c**) ring finger flexion data, (**d**) index and ring flexion data, and (**e**) index and thumb flexion data.

2.2. Data Processing

Figure 3 shows a flowchart of the data processing steps involved in the entire experiment and modeling pipeline. First, the EMG data were recorded by sensors and filtered to eliminate noise. Second, the signals were segmented into shorter sequences, and time and frequency domain features were extracted to capture distinct characteristics of the EMG data. Third, the processed data were used for hand gesture classification using machine learning, where an 80–20 train-test split was performed to train and evaluate the models. Fourth, the hyperparameters of the machine learning models for the given dataset were tuned using a 10-fold grid search cross-validation method to enhance performance. Fifth, the classifiers were trained using the full training dataset and these optimized hyperparameters. Sixth, the model's performance was assessed using unseen testing data and the evaluation metric—the accuracy score—to reflect the model's effectiveness in classifying the EMG data.

Addressing noise in EMG signals is crucial to enhance classification accuracy. Employing an efficient filtering technique significantly contributes to refining EMG signal classification. To refine the EMG signals obtained for specific gestures from each participant, a digital filtering process was applied. This process aimed to eliminate erratic peaks and local extrema clutter from the long-envelope EMG signals as seen closely in Figure 4. Among various filtering methods, the Gaussian smoothing filter (GSF), recommended in [20], emerged as a promising approach, leading to improved EMG signal modeling and accuracy (Figure 4). To facilitate feature extraction-based analysis, the long EMG data sequence per participant for a single gesture was segmented into smaller windows, each containing four gestures for a specific participant. The individual sequences are about

8000 samples long and were stored separately for subsequent time and frequency domain feature extraction.

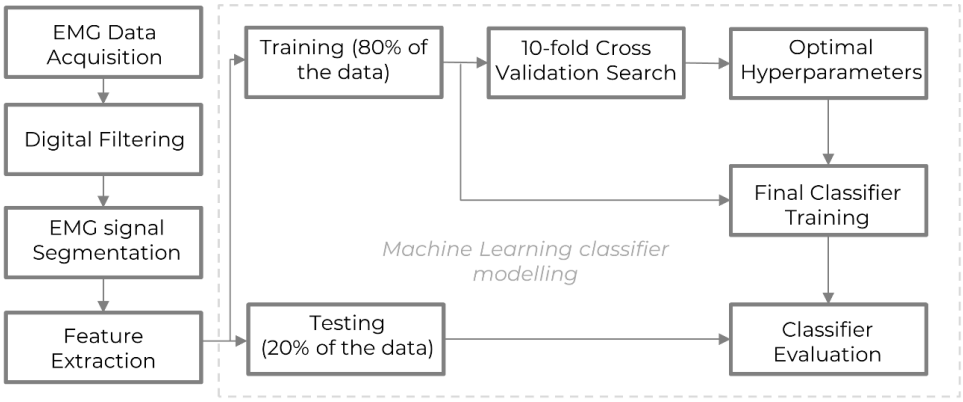


Figure 3. EMG data processing flowchart.

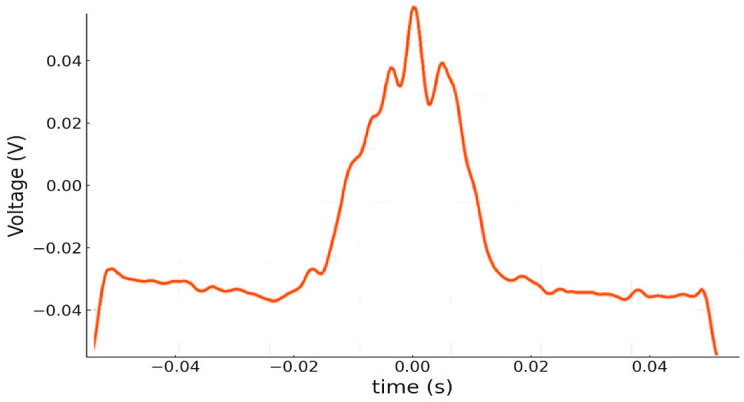


Figure 4. Filtered EMG signal for the power grasp gesture (example shown for configuration C_3 and the power grasp gesture).

2.3. Feature Extraction and Selection

Feature extraction is crucial for reducing the dimensionality of the data representing a gesture, thereby enabling the classification of gestures. A total of 22 time and frequency domain features were extracted from the 8000 sample long segmented EMG signals for every gesture. Features included statistical measures such as root mean square (RMS), variance (VAR), mean absolute value (MAV), slope sign change (SSC), zero crossings (ZC), waveform length (WL), median absolute deviation (MAD), skewness, kurtosis, and energy. These features are well-established in the literature for their utility in gesture recognition [21,22]. The time-frequency domain features include autoregressive model coefficients (5th order), entropy (based on approximate coefficients of 1D discrete wavelet transform of level 4), variance estimates (based on maximum overlap discrete wavelet transform of level 3), spectral entropy, mean frequency, and band power. The wavelet transforms were based on the Daubechies-2 wavelet. The above features have been used multiple times in the literature [10,23,24]. To facilitate effective classification, the extracted features were normalized to a mean of zero by subtracting the mean from every sample and normalizing it using standard deviation.

Principal component analysis (PCA) was utilized in the feature selection phase as a key step to further reduce the dimensionality of the feature space, which was derived from the segmented EMG data obtained by the two sensors. With an initial feature set comprising 44 distinct features (22 $\times$ 2 sensors), PCA was utilized to transform these features into a new set of principal components. The primary objective was to retain the most informative components while reducing dimensionality. The top features capturing over 95% variability

across the dataset were meticulously selected. This selection criterion ensured the retention of only those components that significantly contributed to the data's variance, thereby compressing the feature space while preserving its essential information. These selected components, which captured the most of the dataset's variability, were then exclusively fed into the classifier. This strategic use of reduced, yet informative, PCA-derived features aimed to enhance classification performance by focusing on the most critical aspects of the EMG data.

2.4. Machine Learning Classifier

In this paper, SVM was employed to classify five hand gestures. As SVM is a kernel-based method, selecting proper kernel functions and associated hyper-parameters is an important task. This problem is usually solved by a trial-and-error approach. Moreover, a typical single-kernel SVM application frequently adopts the same hyper-parameters for each class, and it may not be suitable when feature pattern distributions are significantly different among different classes. Although there are different kernels, such as the Gaussian kernel, polynomial kernel, and sigmoid kernel, it is often unclear which is the most suitable kernel for a given dataset, and thus it is desirable for the kernel methods to use an optimized kernel function that adapts well to the data set at hand and the type of the boundaries between classes. An efficient way to design a kernel that is optimal for a given data set is to consider the kernel as a convex combination of basis kernels as illustrated in Equation (1). Such an MKL-based SVM is inspired by [25].

$$K(x, y) = a \cdot K_1(x, y) + b \cdot K_2(x, y) + c \cdot K_3(x, y) + d \cdot KRBF(x, y) \quad (1)$$

Here, K_1 is the linear kernel, K_2 is the quadratic kernel, K_3 is the cubic kernel, and K_{RBF} is the Gaussian or the radial basis function kernel with unity standard deviation. The coefficients $\{a, b, c, d\}$ are hyperparameters to be tuned using a 10-fold grid search cross-validation. The elasticity parameter or box constraint C is also a hyperparameter expressing the degree of loosing constraint. A large C can classify the training samples more correctly but also end up overfitting and reducing the testing accuracy, hence, the box constraint is also tuned using a 10-fold grid search cross-validation. The training and testing of the data are done using a typical 80–20 train-test split. PCA was used for feature selection to further reduce the dimensionality of the input vector given to the classifier and only choose statistically significant features.

Further analysis encompassed utilizing three machine learning classifiers: naïve Bayes, decision tree, and KNN. This was undertaken to evaluate the performance of SVM within the specific dataset and experimental conditions, focusing on robustness with an increasing number of gestures and varying sensor configurations. Although SVMs are well-established as effective for EMG-based gesture recognition [26,27], the research aims to investigate these findings within our unique setup that uses commercially available low-cost sEMG sensors. By confirming the robustness and effectiveness of SVMs in this application, further evidence will be added to the established theory, considering any nuances from our specific dataset.

3. Results

3.1. Sensor Configuration Assessment

The best set of hyperparameters resulting in the highest accuracies on testing data for the SVM, naïve Bayes, KNN, and decision tree classifiers, built using an amalgamated dataset of all participants across different sensor configurations, is shown in Table 1, Table 2, Table 3 and Table 4, respectively. The associated hyperparameters are tuned using a 10-fold grid search cross-validation, and the labels used for each gesture are given in Figure 1.

Table 1. Optimal hyperparameter configuration for SVM-based gesture classification using 10-fold cross-validation. SVM kernel order is the order of pure polynomial kernel. {*a*, *b*, *c*, *d*} are the MKL coefficients. *C* is the box constraint. C_1, C_2, C_3, C_4, C_5 are the five different sensor configurations.

	Hyper Parameters		**Configurations of Sensors**				
			C_1	C_2	C_3	C_4	C_5
Classifying five gestures including coarse pinch, fine pinch, hook grip, point gesture, and power grasp	SVM kernel order		3	-	-	-	-
	MKL coefficients	*a*	-	10	0.1	0.1	1
		b	-	0.5	1	1	0.001
		c	-	0.1	10	10	0
		d	-	0	0	0	0
	C (box constraint)		1	0.1	0.5	0.5	1
Classifying two gestures including fine pinch and power grasp	SVM kernel order		1	3	1	2	3
	MKL coefficients	*a*	-	-	-	-	-
		b	-	-	-	-	-
		c	-	-	-	-	-
		d	-	-	-	-	-
	C (box constraint)		10	1	0.5	0.02	1

Table 2. Optimal hyperparameter configuration for the naïve Bayes-based gesture classification using 10-fold cross-validation. Distribution kernel 'N' is the normal kernel and 'B' is the box (uniform) kernel.

	Hyper Parameters	**Configurations of Sensors**				
		C_1	C_2	C_3	C_4	C_5
Classifying five gestures including coarse pinch, fine pinch, hook grip, point gesture, and power grasp	Distribution kernel (N/B)	N	B	N	N	N
	Bandwidth	0.05	0.8	0.08	0.15	0.42
Classifying two gestures including fine pinch and power grasp	Distribution kernel (N/B)	N	N	N	N	N
	Bandwidth	0.1	0.05	0.1	0.05	0.05

Table 3. Optimal hyperparameter configuration for the KNN-based gesture classification using 10-fold cross-validation. D_1 to D_4 denote Euclidean, cosine, city block, and Minkowski distance functions.

	Hyper Parameters	**Configurations of Sensors**				
		C_1	C_2	C_3	C_4	C_5
Classifying five gestures including coarse pinch, fine pinch, hook grip, point gesture, and power grasp	Distance function	D_1	D_2	D_1	D_1	D_1
	K value	1	1	1	1	2
Classifying two gestures including fine pinch and power grasp	Distance function	D_1	D_3	D_2	D_1	D_4
	K value	1	1	1	1	1

Table 4. Optimal hyperparameter configuration for the decision tree-based gesture classification using 10-fold cross-validation.

	Hyper Parameters	**Configurations of Sensors**				
		C_1	C_2	C_3	C_4	C_5
Classifying five gestures including coarse pinch, fine pinch, hook grip, point gesture, and power grasp	Minimum leaf size	1	9	2	2	2
Classifying two gestures including fine pinch and power grasp	Minimum leaf size	2	9	9	2	2

Table 5 presents the performance results of various classifiers on two separate gesture classification tasks with different configurations (C_1, C_2, C_3, C_4, C_5) and provides a comparative overview of the accuracy of different classifiers across five distinct and unrelated sensor configurations for gesture classification tasks. In classifying five gestures, the SVM classifier's performance ranges from 75% to 90%, with the highest accuracy observed in configuration C_5. The KNN's accuracy fluctuates, achieving its highest at 82% with C_4 and slightly lower at 80% with C_5. Naïve Bayes shows its best result at 70% with C_5, while the decision tree classifier caps at 75% with the same configuration.

Table 5. Testing accuracies for the SVM, KNN, naïve Bayes, and decision tree classifiers across different sensor configurations. C_1 to C_5 are the five different sensor configurations.

	Classifier	**Configuration C_i**				
		C_1	C_2	C_3	C_4	C_5
Classifying five gestures including coarse pinch, fine pinch, hook grip, point gesture, and power grasp	SVM	75%	75%	84.6%	87.2%	90%
	KNN	67.9%	73.9%	69.2%	82%	80%
	Naïve Bayes	53.6%	51.3%	61.5%	61.5%	70%
	Decision Tree	57.2%	59%	56.4%	59%	75%
Classifying two gestures including fine pinch and power grasp	SVM	100%	100%	100%	100%	100%
	KNN	100%	86.7%	100%	93.3%	100%
	Naïve Bayes	100%	73.3%	100%	93.3%	93.3%
	Decision Tree	100%	86.7%	100%	80%	100%

In contrast, the task of classifying two gestures, fine pinch and power grasp, sees a notably higher and perfect performance from the SVM classifier, maintaining 100% accuracy across all configurations. The KNN and decision tree classifiers also perform exceptionally well, with KNN achieving 100% accuracy in all but C_2 and C_4, where it scores slightly lower at 86.7% and 93.3%, respectively. The Naïve Bayes and decision tree classifiers exhibit a perfect 100% accuracy for configuration C_3. This distinct difference in performance across tasks suggests that certain classifiers, especially SVM, may be more robust to changes in configurations or are better suited for binary classification tasks in the context of gesture recognition.

3.2. Classifier Assessment

The confusion matrix for the SVM classifier across different configurations has been meticulously plotted to provide a comprehensive visualization of the model's performance (Figure 5).

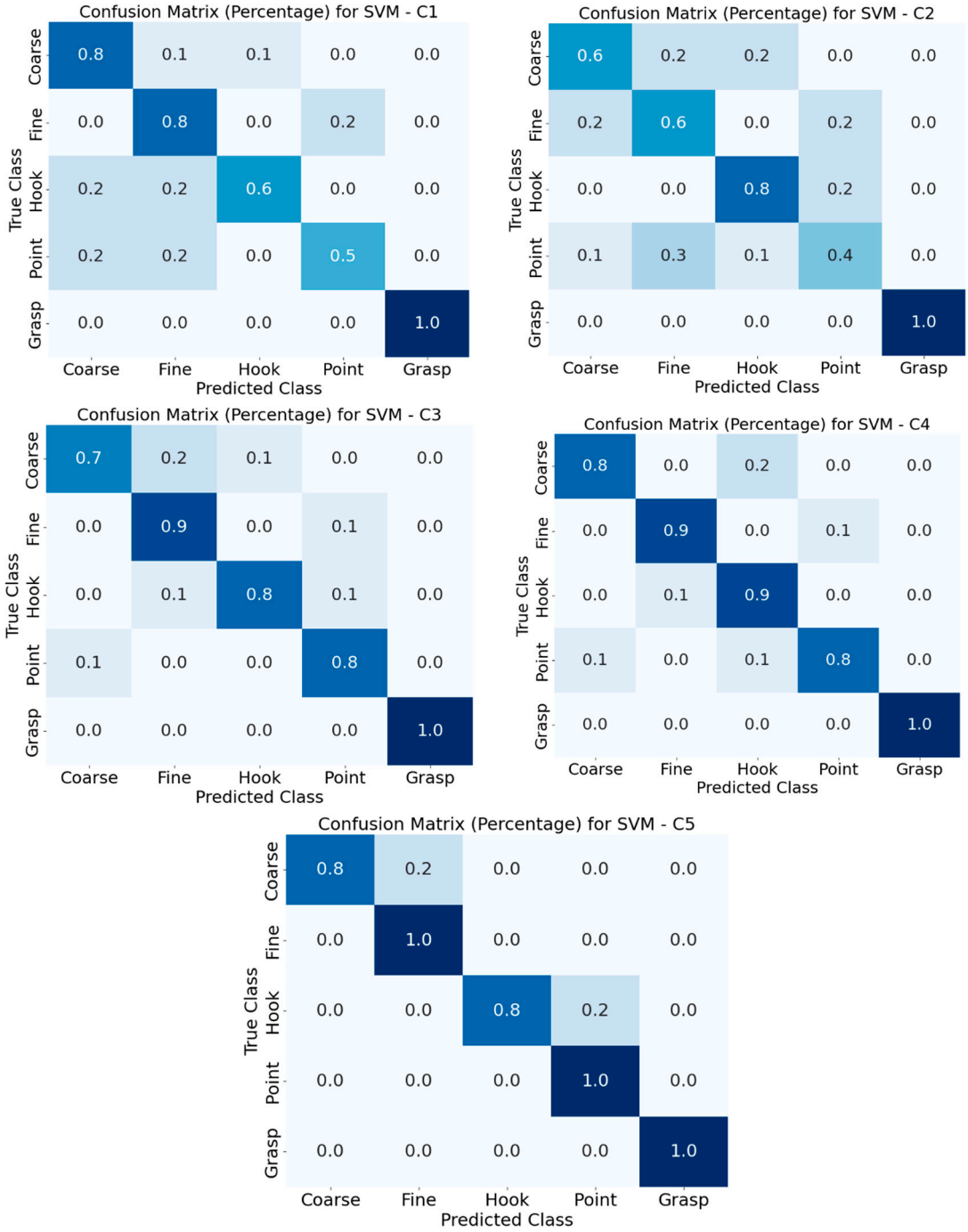


Figure 5. Confusion matrix for SVM classifier across different configurations.

4. Discussion

The study extends beyond assessing classification accuracy, delving into the complexities of classifying an increasing array of gestures, as corroborated by prior research [28]. This investigation further reveals the limitations of using a single sensor for EMG data acquisition, which restricts the number of distinctly identifiable classes. We evaluated the effectiveness of four machine learning classifiers—SVM, naïve Bayes, KNN, and decision tree—over five distinct sensor configurations for collecting detailed hand gesture EMG data from the forearms of eight participants. An extensive set of 22 features, encompassing both time and frequency domains per sensor, were extracted from the EMG signals. Prior studies suggest that a mixed-domain feature set can bolster classifier performance [29].

The application of PCA to select key features significantly reduced feature space dimensionality, enhancing classification accuracy. The results, as summarized in Tables 1–5, identified sensor setup C_5 (two sensors placed proximal to the wrist along flexor digitorum superficialis and flexor pollicis longus) as the most effective, yielding the highest accuracy with SVM, naïve Bayes, and decision tree classifiers. The SVM classifier, empowered by MKL, consistently achieved over 90% accuracy in gesture classification through C_5, as shown in Table 2 and Figure 5. The dual-sensor configuration of C_5, which captures data from the muscles controlling thumb and index finger movements, suggests an advantageous strategy for sensor placement in prosthetic hand design.

The fine pinch gesture, depicted in Figure 1b, emerged as particularly challenging due to its lower muscle contraction levels and reliance on just two fingers, making it more prone to noise. C_5's precise sensor positioning over the active muscles during this gesture enabled the capture of more nuanced data. A comparison of C_1 and C_5 (Figure 2) illustrates the significant influence of an additional EMG sensor on classification accuracy, confirming the superiority of dual-channel over single-channel EMG in gesture recognition.

When space constraints limit sensor placement near the wrist on a prosthetic arm, C_4 (two sensors placed proximal to the wrist along flexor digitorum superficialis and along flexor digitorum superficialis and flexor digitorum profundus) emerges as a feasible alternative, achieving 87.2% accuracy with SVM (Table 5). These results highlight the importance of both sensor location on the muscle and proximity to the muscle belly for optimal data collection.

MKL's enhancement of SVM performance is evident in sensor configuration C_3 (one sensor placed along flexor digitorum superficialis and flexor digitorum profundus), where a single sensor competes closely with the dual-sensor C_4 in accuracy (SVM testing accuracy, Table 5). MKL's adaptive selection of kernel functions for C_3's data is a significant advance over single-kernel methods. The SVM's hyperparameter C plays a pivotal role in balancing model complexity against overfitting, with a range of values from 0.01 to 10, evaluated through a 10-fold grid search cross-validation (Table 1). This fine-tuning was crucial for developing an optimized SVM model with strong generalization capabilities for precise gesture classification. Thus, for prosthetic arms that can only incorporate one EMG sensor, C_3 is the recommended setup. The use of MKL with SVM significantly improves performance over base SVMs, especially for classifying multiple gestures and allowed us to achieve non-zero MKL coefficients for classifying all five gestures as detailed in Table 1. While base SVMs excel in binary classification, MKL handles the complexity of EMG signals better by combining multiple kernels. This results in robust classification of five distinct gestures, justifying the added complexity of MKL.

The confusion matrices demonstrate SVM's high performance over different gestures at around 80% to 100% with the C_5 configuration. Additionally, SVM excelled as a binary classifier for the fine pinch versus power grasp gestures (Figure 1), achieving 100% accuracy. A noted trend is the reduction in classification accuracy with an increasing number of gestures, a phenomenon that resonates with previous findings [26]. This decline is attributed to the broader dispersion of EMG signals in the forearm and the resultant signal overlap from simultaneous muscle contractions when performing complex gestures.

A potential limitation of the present study, when extended to real-time online EMG classification, is the slight variation in the lengths of EMG segments in the time domain used for feature extraction across different gesture configurations. This limitation can be readily addressed by selecting precisely equal segment lengths during online application. Another limitation of the current work is the sample size of eight participants. This sample size is similar to other samples sizes in the literature [25,27]. However, the limited sample size may affect the robustness of accuracies obtained to a certain level. The current study serves to validate the proposed methodology on a small scale, and it is planned to conduct future experiments with a larger sample size and include individuals with limb differences to directly evaluate the application of our findings in prosthetic hand control. This will enhance the generalizability and relevance of the current research.

Future research will apply the findings from this paper to replicate hand gestures in a myoelectric hand prosthetic in real-time. It will also explore different sensor configurations and machine learning techniques to further enhance the performance of myoelectric prostheses. Additionally, the development of more sophisticated feature extraction and classifier optimization methods could be beneficial in handling multifaceted EMG signal analysis. This study serves as a steppingstone towards the realization of more efficient and effective myoelectric prostheses, and it is hoped that the insights gained will inspire further exploration in this promising field.

5. Conclusions

This study has made significant strides in the field of myoelectric prosthetic technology, with key findings that have the potential to shape future research and applications. The superior performance of the dual-channel configuration index-ring and index-thumb (C_4 and C_5) underscores the importance of optimal sensor placement in enhancing the functionality of myoelectric prostheses. The application of MKL in the SVM classifier, particularly in placing one sensor in ring (configuration C_3), has demonstrated its efficacy in achieving high classification accuracy even with a single sensor. The observed decrease in accuracy with an increase in the number of gestures highlights the need for comprehensive feature extraction and classifier optimization for complex EMG signal analysis. These findings collectively pave the way for advancements in myoelectric prostheses, emphasizing the need for refined sensor configurations and machine learning methodologies to improve gesture recognition accuracy.

Author Contributions: Conceptualization, L.C., S.H., K.B.M., J.A. and H.R.; Methodology, L.C., S.H., K.B.M., J.A. and H.R.; Software, L.C. and S.H.; Validation, L.C., S.H., K.B.M. and H.R.; Formal analysis, L.C., S.H., J.A. and H.R.; Investigation, L.C., S.H., K.B.M., J.A. and H.R.; Resources, K.B.M., J.A. and H.R.; Data curation, L.C., S.H., K.B.M. and H.R.; Writing—original draft, L.C., S.H., K.B.M., J.A. and H.R.; Writing—review & editing, L.C., S.H., K.B.M., J.A. and H.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Glenrose Hospital Foundation and Mitacs, grant number: IT36690.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki and approved by the Institutional Ethics Committee of the University of Alberta (AB T6G 2N2, approved on 25 April 2022).

Informed Consent Statement: All participants signed a consent form, the procedure was approved by the university of Alberta research ethics board application number: Pro00117863.

Data Availability Statement: Data available upon request.

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

References

1. McDonald, C.L.; Westcott-McCoy, S.; Weaver, M.R.; Haagsma, J.; Kartin, D. Global prevalence of traumatic non-fatal limb amputation. *Prosthet. Orthot. Int.* **2021**, *45*, 105–114. [CrossRef] [PubMed]
2. Efanov, J.; Tchiloemba, B.; Izadpanah, A.; Harris, P.; Danino, M. A review of utilities and costs of treating upper extremity amputations with vascularized composite allotransplantation versus myoelectric prostheses in Canada. *JPRAS Open* **2022**, *32*, 150–160. [CrossRef] [PubMed]
3. Hero Arm User Guide—Open Bionics. Available online: https://openbionics.com/hero-arm-user-guide/ (accessed on 9 December 2023).
4. bebionic | Ottobock US. Available online: https://www.ottobockus.com/prosthetics/upper-limb-prosthetics/solution-overview/bebionic-hand/ (accessed on 9 December 2023).
5. Peerdeman, B.; Boere, D.; Witteveen, H.; Hermens, H.; Stramigioli, S.; Rietman, H.; Veltink, P.; Misra, S. Myoelectric forearm prostheses: State of the art from a user-centered perspective. *J. Rehabil. Res. Dev.* **2011**, *48*, 719–738. [CrossRef] [PubMed]
6. Cordella, F.; Ciancio, A.L.; Sacchetti, R.; Davalli, A.; Cutti, A.G.; Guglielmelli, E.; Zollo, L. Literature review on needs of upper limb prosthesis users. *Front. Neurosci.* **2016**, *10*, 209. [CrossRef] [PubMed]
7. Geethanjali, P. Myoelectric control of prosthetic hands: State-of-the-art review. *Med. Devices* **2016**, *9*, 247–255. [CrossRef] [PubMed]

8. Tavakoli, M.; Benussi, C.; Lopes, P.A.; Osorio, L.B.; de Almeida, A.T. Robust hand gesture recognition with a double channel surface EMG wearable armband and SVM classifier. *Biomed. Signal Process. Control.* **2018**, *46*, 121–130. [CrossRef]
9. Palkowski, A.; Redlarski, G. Basic Hand Gestures Classification Based on Surface Electromyography. *Comput. Math. Methods Med.* **2016**, *2016*, 6481282. [CrossRef] [PubMed]
10. Lee, K.H.; Min, J.Y.; Byun, S. Electromyogram-based classification of hand and finger gestures using artificial neural networks. *Sensors* **2021**, *22*, 225. [CrossRef] [PubMed]
11. MAhsan, R.; Ibrahimy, M.I.; Khalifa, O.O. Electromygraphy (EMG) signal based hand gesture recognition using artificial neural network (ANN). In Proceedings of the 2011 4th International Conference on Mechatronics: Integrated Engineering for Industrial and Societal Development, ICOM'11—Conference Proceedings, Kuala Lumpur, Malaysia, 17–19 May 2011. [CrossRef]
12. Kisa, D.H.; Ozdemir, M.A.; Guren, O.; Akan, A. EMG based Hand Gesture Classification using Empirical Mode Decomposition Time-Series and Deep Learning. In Proceedings of the TIPTEKNO 2020—Tip Teknolojileri Kongresi—2020 Medical Technologies Congress, TIPTEKNO 2020, Antalya, Turkey, 19–20 November 2020. [CrossRef]
13. MYOWARE® 2.0 Muscle Sensor. Available online: https://myoware.com/products/muscle-sensor/ (accessed on 9 December 2023).
14. del Toro, S.F.; Wei, Y.; Olmeda, E.; Ren, L.; Guowu, W.; Díaz, V. Validation of a Low-Cost Electromyography (EMG) System via a Commercial and Accurate EMG Device: Pilot Study. *Sensors* **2019**, *19*, 5214. [CrossRef] [PubMed]
15. Heywood, S.; Pua, Y.H.; McClelland, J.; Geigle, P.; Rahmann, A.; Bower, K.; Clark, R. Low-cost electromyography–Validation against a commercial system using both manual and automated activation timing thresholds. *J. Electromyogr. Kinesiol.* **2018**, *42*, 74–80. [CrossRef] [PubMed]
16. Kurniawan, S.R.; Pamungkas, D. MYO Armband sensors and Neural Network Algorithm for Controlling Hand Robot. In Proceedings of the 2018 International Conference on Applied Engineering, ICAE 2018, Batam, Indonesia, 3–4 October 2018. [CrossRef]
17. Mao, Z.-H.; Lee, H.-N.; Sclabassi, R.J.; Sun, M. Information capacity of the thumb and the index finger in communication. *IEEE Trans. Biomed. Eng.* **2009**, *56*, 1535–1545. [CrossRef] [PubMed]
18. Hioki, M.; Kawasaki, H. Estimation of Finger Joint Angles from sEMG Using a Neural Network Including Time Delay Factor and Recurrent Structure. *ISRN Rehabil.* **2012**, *2012*, 604314. [CrossRef]
19. Experiment: Signal Classification. Available online: https://backyardbrains.com/experiments/RobotHand (accessed on 9 December 2023).
20. Ghalyan, I.F.J.; Abouelenin, Z.M.; Annamalai, G.; Kapila, V. Gaussian Smoothing Filter for Improved EMG Signal Modeling. In *Signal Processing in Medicine and Biology: Emerging Trends in Research and Applications*; Springer: Cham, Switzerland, 2020; pp. 161–204. [CrossRef]
21. Jaramillo-Yánez, A.; Benalcázar, M.E.; Mena-Maldonado, E. Real-Time Hand Gesture Recognition Using Surface Electromyography and Machine Learning: A Systematic Literature Review. *Sensors* **2020**, *20*, 2467. [CrossRef] [PubMed]
22. Tkach, D.; Huang, H.; Kuiken, T.A. Study of stability of time-domain features for electromyographic pattern recognition. *J. Neuroeng. Rehabil.* **2010**, *7*, 21. [CrossRef] [PubMed]
23. Khairuddin, I.M.; Sidek, S.N.; Majeed, A.P.P.A.; Razman, M.A.M.; Puzi, A.A.; Yusof, H.M. The classification of movement intention through machine learning models: The identification of significant time-domain EMG features. *PeerJ Comput. Sci.* **2021**, *7*, e379. [CrossRef] [PubMed]
24. Tomaszewski, J.; Amaral, T.; Dias, O.; Wołczowski, A.; Kurzyński, M. EMG signal classification using neural network with AR model coefficients. *IFAC Proc. Vol.* **2009**, *42*, 318–325. [CrossRef]
25. She, Q.; Luo, Z.; Meng, M.; Xu, P. Multiple kernel learning SVM-based EMG pattern classification for lower limb control. In Proceedings of the 11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010, Singapore, 7–10 December 2010; pp. 2109–2113. [CrossRef]
26. Odeyemi, J.; Ogbeyemi, A.; Wong, K.; Zhang, W. On Automated Object Grasping for Intelligent Prosthetic Hands Using Machine Learning. *Bioengineering* **2024**, *11*, 108. [CrossRef] [PubMed]
27. Aviles, M.; Sánchez-Reyes, L.-M.; Fuentes-Aguilar, R.Q.; Toledo-Pérez, D.C.; Rodríguez-Reséndiz, J. A Novel Methodology for Classifying EMG Movements Based on SVM and Genetic Algorithms. *Micromachines* **2022**, *13*, 2108. [CrossRef] [PubMed]
28. Fajardo, J.M.; Gomez, O.; Prieto, F. EMG hand gesture classification using handcrafted and deep features. *Biomed. Signal Process. Control.* **2021**, *63*, 102210. [CrossRef]
29. Abbaspour, S.; Lindén, M.; Gholamhosseini, H.; Naber, A.; Ortiz-Catalan, M. Evaluation of surface EMG-based recognition algorithms for decoding hand movements. *Med. Biol. Eng. Comput.* **2020**, *58*, 83–100. [CrossRef] [PubMed]

electronics

MDPI

Article

Improving Data Utility in Privacy-Preserving Location Data Collection via Adaptive Grid Partitioning

Jongwook Kim

Department of Computer Science, Sangmyung University, Seoul 03016, Republic of Korea; jkim@smu.ac.kr

Abstract: The widespread availability of GPS-enabled devices and advances in positioning technologies have significantly facilitated collecting user location data, making it an invaluable asset across various industries. As a result, there is an increasing demand for the collection and sharing of these data. Given the sensitive nature of user location information, considerable efforts have been made to ensure privacy, with differential privacy (DP)-based schemes emerging as the most preferred approach. However, these methods typically represent user locations on uniformly partitioned grids, which often do not accurately reflect the true distribution of users within a space. Therefore, in this paper, we introduce a novel method that adaptively adjusts the grid in real-time during data collection, thereby representing users on these dynamically partitioned grids to enhance the utility of the collected data. Specifically, our method directly captures user distribution during the data collection process, eliminating the need to rely on pre-existing user distribution data. Experimental results with real datasets show that the proposed scheme significantly enhances the utility of the collected location data compared to the existing method.

Keywords: location privacy; density distribution; differential privacy; geo-indistinguishability

Citation: Kim, J. Improving Data Utility in Privacy-Preserving Location Data Collection via Adaptive Grid Partitioning. *Electronics* **2024**, *13*, 3073. https://doi.org/10.3390/electronics13153073

Academic Editors: Swapnoneel Roy and Guosheng Xu

Received: 27 June 2024
Revised: 27 July 2024
Accepted: 29 July 2024
Published: 3 August 2024

1. Introduction

The proliferation of GPS-enabled devices and recent advances in positioning technologies have made it easier to collect user location data, making them a valuable asset for various sectors. These data play an important role in areas such as personalized marketing, real-time traffic analysis, recommendations, etc. For example, real-time traffic analysis utilizes location data to optimize traffic flow, reduce congestion, and improve navigation systems for more efficient travel [1,2]. Additionally, location-based recommendations for services, restaurants, and events provide users with relevant and timely suggestions, enhancing their overall experience [3,4]. As a result, the demand for collecting and sharing user location data continues to increase.

User location data are sensitive because they contain personal information, such as home or company addresses, hospital visit records, and even political affiliations [5–7]. For example, by collecting and analyzing the positioning information of visitors in a large indoor shopping mall, it is possible to infer sensitive details, such as their shopping patterns. In addition, location data can be cross-referenced with other data sets to draw even more precise conclusions about an individual's lifestyle and choices [8]. For example, frequent visits to certain types of businesses or locations can indicate specific health conditions, hobbies or even religious practices. As a result, the indiscriminate collection of location data raises significant privacy concerns. Consequently, considerable efforts have been made to protect the privacy of users' location data when handling such data.

As differential privacy (DP) [9,10] has become the de facto standard for handling sensitive personal data, significant efforts have been made to apply it to location data. As a result, numerous DP-based methods have been proposed to collect, process, and analyze location data while preserving privacy. Many of these approaches represent user location data using grids, where the entire domain is uniformly partitioned into disjoint

grids, and a user's location is represented by the grid in which his or her actual position lies [11–14]. Although representing user location using uniformly partitioned grids is straightforward, it does not account for the actual distribution of users within the space. This approach often results in lower utility of collected location data, as it assumes that users are evenly and uniformly distributed throughout the area. However, this is not true in most real-world scenarios, where some areas are denser than others. For example, in an urban environment, the city center may have a high concentration of users, while the suburbs have a lower density. This discrepancy can negatively impact the accuracy and effectiveness of subsequent analyses. Therefore, more sophisticated grid representations that align with actual user distributions are necessary to improve data utilization and enhance the performance of these applications.

Existing solutions assume the existence of prior information about the distribution of users, typically obtained from historical data. However, such historical data may not always be available for many applications. More importantly, prior information about the distribution of users derived from historical data may not match the current distribution as it may change over time or in response to special social events. For example, major events such as festivals can dramatically change user movement patterns and densities, rendering historical data obsolete or misleading [15]. Therefore, it is preferable to instantaneously extract information about the distribution of users during data collection and adaptively adjust the grid accordingly.

In this paper, we propose a novel method that simultaneously extracts the distribution of users and adaptively adjusts the grid in real-time during location data collection. The contributions of this work can be summarized as follows:

- First, we introduce a method to effectively compute the distribution of users during DP-based location data collection. This approach is able to effectively capture user distribution in real-time and adapt to dynamic changes in user behavior.
- Then, we propose a method to adaptively adjust the grid to maximize the utility of the collected location data under DP. This adaptive grid adjustment is designed to improve the granularity and relevance of the data, ensuring that the most significant and densely populated areas are prioritized, thus improving the overall quality and applicability of the data.
- We evaluated the performance of the proposed algorithms using real-world datasets. The evaluation results demonstrated that the proposed scheme significantly enhances the utility of the collected location data compared to existing methods

The rest of this paper is organized as follows: Section 2 reviews related work. In Section 3, we provide background information. In Section 4, we introduce a novel method that simultaneously extracts user distribution and adaptively adjusts the grid in real-time during location data collection. In Section 5, we experimentally evaluate the proposed approach with real datasets. Finally, Section 6 presents our conclusions

2. Related Work

Numerous DP-based methods have been developed to collect, process, and analyze location data while preserving privacy. In this section, we provide a brief overview of these methods.

Local differential privacy (LDP) is a variant of DP in which each user individually perturbs his own sensitive data before reporting it to the server. Kim and Jang [12] propose an LDP-based data aggregation approach designed for workload-aware collection of indoor positioning data, while ensuring user privacy. Their method identifies an optimal data encoding and perturbation strategy within the LDP framework to minimize the overall estimation error for the given workload. LDPTrace [14] is designed to synthetically generate locally differentially private trajectory data. In this method, user location information is collected using LDP to ensure privacy, and these perturbed data are then used to generate synthetic trajectories. Kim et al. [3] present a method for recommending the next point-of-interest, utilizing location data collected under LDP.

Metric differential privacy (MDP) extends the standard differential privacy framework to handle data with inherent metric or distance measures [16]. This extension is particularly useful for location-based data. Geo-indistinguishability (Geo-Ind) is a specific application of MDP designed for location-based services [11,17,18]. Mobile Crowdsensing (MCS) frameworks often use Geo-Ind to collect location information from workers and assign tasks in a privacy-preserving manner. Wang et al. [19] was the first to use Geo-Ind to protect the location privacy of workers in the MCS process. Their proposed framework includes three steps: First, the MCS server generates a function that satisfies Geo-Ind. Next, each worker downloads this function, obfuscates their true location, and uploads the obfuscated location to the server. Finally, the MCS server assigns tasks to workers based on the obfuscated location information. In [20], location privacy protection in vehicle-based MCS is investigated, where the roadmap is modeled as a weighted directed graph with task and worker locations as points on the graph. The authors propose an optimization mechanism-based obfuscation scheme that achieves location obfuscation through a probabilistic distribution over the graph that satisfies Geo-Ind. Jin et al. [21] proposes a user-centric location privacy trading framework for MCS. Following the notion of Geo-Ind, they design a location obfuscation mechanism that allows each worker to probabilistically obfuscate his true location using his own privacy budget. Zhang et al. [13] introduces an obfuscation method that satisfies Geo-Ind to collect location information from workers in MCS. Huang et al. [22] propose a privacy-aware scheme for MCS-based noise monitoring, where the server publishes tasks and workers report perturbed locations and noise levels under DP. Each worker collaborates with a master, carefully selected from the workers in the same group, to achieve group-level Geo-Ind. Zhao et al. [23] explored the privacy protection of individuals' locations in the context of analyzing the geographic directional distribution of the community. They defined community information using a covariance matrix and integrated it into the proposed geo-ellipse indistinguishability based on Geo-Ind. This geo-ellipse indistinguishability provides quantifiable privacy guarantees for locations within Mahalanobis space. Yu et al. [24] highlighted the weaknesses of current Geo-Ind-based location obfuscation mechanisms, especially when users consistently share their locations with multiple LBS providers over a long period of time. To address this issue, they introduced PrivLocAd, a system that uses location profiling to generate obfuscated locations, thereby protecting user privacy against multi-platform adversarial attacks. Zhao et al. [25] introduced a novel privacy concept called vector-indistinguishability, which builds on Geo-Ind to provide a privacy guarantee for location-dependent relations. They have developed four mechanisms to achieve vector-indistinguishability, using both Laplace and uniform distributions. Mendes et al. [26] utilized user velocity and report frequency to measure the correlation between locations. They extended Geo-Ind to enhance privacy preservation in continuous online reporting scenarios. Specifically, they introduced a velocity-aware Geo-Ind that automatically balances privacy and utility based on the user's velocity and frequency of location reports.

EGeoIndis [27] is a vehicle location privacy protection framework designed for traffic density estimation. It leverages Geo-Ind to protect vehicle location privacy during the traffic density estimation process. In [28], the authors proposed a deep learning-based method to estimate the density distribution using location data collected under Geo-Ind. Chen et al. [29] develop a method to create a COVID-19 vulnerability map using the density distribution of volunteer participants with COVID-19 symptoms. They exploit Geo-Ind to collect participants' locations in a privacy-preserving manner to ensure the confidentiality of sensitive health information. Fathalizadeh et al. [30] present a framework for implementing Geo-Ind for indoor environments. The proposed framework considers two scenarios for applying Geo-Ind, reporting an obfuscated point to the location service provider that satisfies DP.

The proposed approach in this paper leverages Geo-Ind, which is a representative model in the domain of privacy-preserving location data collection. However, the proposed approach differs from other Geo-Ind-based methods in several ways. First, existing

methods typically rely on the availability of historical data to infer user distribution, which presents significant challenges. Historical data may not always be accessible or current, resulting in inaccurate inferences of user distribution. Second, most existing methods use static grid structures, resulting in a fixed representation that does not account for the dynamic movement of users. In contrast, the proposed method addresses these limitations by adaptively adjusting grids in real-time during data collection. This dynamic adjustment captures the current user distribution without relying on historical data, thereby enhancing the utility of the collected data.

3. Background and Problem Statement

In this section, we provide the necessary background for this paper and state the problem addressed in this paper.

3.1. Background

Recently, DP has emerged as the de facto standard for privacy-preserving data processing. DP is based on a formal mathematical definition that provides a probabilistic privacy guarantee against attackers with arbitrary background knowledge [9]. It ensures that an attacker cannot determine with high confidence whether a given individual is included in the disseminated data. DP is formally defined as [9,10]:

Definition 1. *(ϵ-DP) A randomized algorithm $\mathcal{A}$ satisfies ϵ-DP, if and only if for (1) any two neighboring datasets, D_1 and D_2, and (2) any output O of $\mathcal{A}$, the following is satisfied:*

$$Pr[\mathcal{A}(D_1) = O] \leq e^{\epsilon} \times Pr[\mathcal{A}(D_2) = O]. \tag{1}$$

Two datasets, D_1 and D_2, are considered neighboring if they differ by only one record. The above definition indicates that, for any output of $\mathcal{A}$, an adversary with any amount of background knowledge cannot reliably determine whether D_1 or D_2 was the input to $\mathcal{A}$. The parameter ϵ, known as the privacy budget, regulates the privacy level: smaller ϵ values provide stronger privacy protection but add more noise to the result, whereas larger ϵ values offer weaker privacy protection with less noise.

There have been several proposals to apply the concept of DP to the protection of location data. In this paper, we use Geo-Ind, a concept based on the well-established DP framework and recognized as the standard privacy definition for protecting location data in location-based services [11,17,18]. In addition to location data, Geo-Ind is also used to collect other types of data, such as text microdata, in a DP-compliant manner [31,32]. Geo-Ind is formally defined as follows:

Definition 2. *(ϵ-Geo-Ind) Consider $\mathcal{X}$ as the set of possible user locations and $\mathcal{Y}$ as the set of reported locations, which are typically assumed to be equal. Let K be a randomized mechanism that generates a perturbed location from a user's true location. A randomized mechanism K satisfies ϵ-Geo-Ind if and only if the following condition holds for (1) all $x_1, x_2 \in \mathcal{X}$ and (2) any output location $y \in \mathcal{Y}$:*

$$K(x_1)(y) \leq e^{\epsilon \cdot d(x_1, x_2)} \times K(x_2)(y), \tag{2}$$

where $d(x_1, x_2)$ corresponds to the distance between x_1 and x_2.

There are two primary methods for implementing Geo-Ind: the Laplace mechanism and the matrix-based mechanism. It is well known that the matrix-based mechanism is more effective than the Laplace method, given prior information about the distribution of users which can be obtained from available historical data [11]. This increased effectiveness is due to the fact that the matrix-based mechanism incorporates prior distribution information when perturbing the true locations of users. As a result, the distribution of perturbed locations collected using the matrix-based mechanism more closely approximates the true distribution than the distribution collected using the Laplace mechanism.

In the matrix-based mechanism, the space is first partitioned into a set of grids, and then the data collection server computes an obfuscation matrix, M, over these grids that satisfy ϵ-Geo-Ind. This matrix is then distributed to users. Subsequently, users perturb their location data according to the probabilities embedded in M and report the perturbed location to the server instead of their true data. Several approaches for computing the obfuscation matrix that satisfies ϵ-Geo-Ind have been proposed in the literature [11,13,14,32,33]. We, however, note that the method proposed in this paper is general enough to be applied to any matrix-based mechanism.

3.2. Problem Statement

Let $U = \{u_1, u_2, \cdots, u_k\}$ be a set of users who agree to provide their location information to the server. However, users do not fully trust the server and thus, instead of providing true location information, each user provides perturbed (and thus privacy-preserved) location information that satisfies ϵ-Geo-Ind. Let us assume that the entire area is divided into disjoint grids, and let G be the set of these grids. Each user's location is then represented by the grid in G to which his/her true location belongs.

The problem addressed in this paper is to collect high-utility location data while protecting users' location privacy with ϵ-Geo-Ind. Existing methods either use static grid partitioning that does not adapt to real-time changes in user distribution or rely on pre-existing user distribution data, which may not be available or accurate in real-time scenarios. In order to address these gaps, we propose a novel adaptive grid partitioning method that dynamically adjusts the grid during the process of location data collection. In particular, the proposed method directly captures user distribution during data collection, eliminating the need for pre-existing distribution information.

4. Proposed Method

Figure 1 provides an overview of the proposed location data collection scheme using ϵ-Geo-Ind.

- Collection of perturbed location data from sampled users: The server first computes the obfuscation matrix, M, over uniformly partitioned grids and distributes it to the sampled users, who then send their perturbed locations back to the server.
- Estimation of the distribution of users: The server estimates the distribution of users based on the perturbed location data collected from the sampled users.
- Computation of adaptively partitioned grids: The server uses the estimated distribution of users to compute adaptively partitioned grids.
- Collection of perturbed location data from remaining users using adaptively partitioned grids: A new obfuscation matrix is computed using the adaptively partitioned grids. This new obfuscation matrix is then used to collect location data from the remaining users.

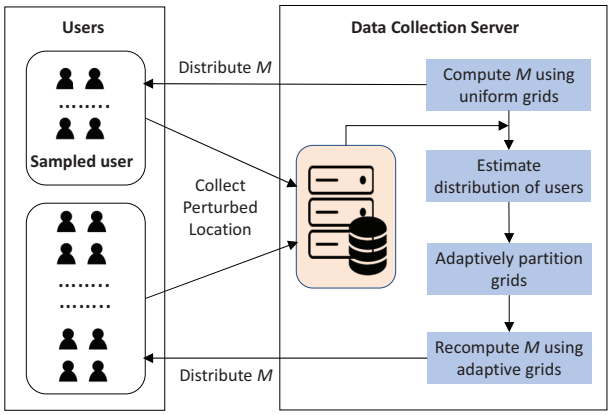


Figure 1. An overview of the proposed privacy-preserving location data collection scheme.

In next subsections, we provide a detailed explanation of each step.

4.1. Collecting Perturbed Location Data from Sampled Users

Let us assume that the entire area is uniformly partitioned into m grids, $G = \{g_1, g_2, \cdots, g_m\}$. The data collection server computes an obfuscation matrix, M, over G. There are various approaches for computing M that satisfy ϵ-Geo-Ind. In this paper, we use the method proposed in [13], where the obfuscation matrix is defined as an $m \times m$ matrix. Each element $M[i, j]$, which represents the probability that a perturbed location g_j is randomly generated from the true location g_i, is defined as follows:

$$M[i, j] = \frac{e^{-\frac{\epsilon}{2} \cdot d(g_i, g_j)}}{\sum_{g_k \in G} e^{-\frac{\epsilon}{2} \cdot d(g_i, g_k)}} \tag{3}$$

Once computing the obfuscation matrix M, it is distributed to the sampled users, who then perturb their true location according to the probabilities encoded in M and report the perturbed location to the server.

4.2. Estimating Probability Distribution

After collecting perturbed location data from sampled users, the next step is to estimate the distribution of users based on these data. For each grid $g_i \in G$, let $P(g_i)$ be the probability that a user is located at g_i. Then, in this subsection, we estimate $P(g_i)$ for all $g_i \in G$ from the sampled perturbed location data.

Let $g'_j \in G$ be the perturbed data that the server receives from a sampled user. For the sake of explanation, we will use g'_j to denote the perturbed location and g_j to denote the true location. The probability that this perturbed location is randomly generated from the true location $g_i \in G$ can be computed as follows:

$$P(g_i|g'_j) = \frac{P(g_i)P(g'_j|g_i)}{p(g'_j)} = \frac{P(g_i)P(g'_j|g_i)}{\sum_{g_k \in G} P(g_k)P(g'_j|g_k)} = \frac{P(g_i)M[i, j]}{\sum_{g_k \in G} P(g_k)M[k, j]} \tag{4}$$

Note, that $P(g'_j|g_i) = M[i, j]$ by the definition of the obfuscation matrix. Since it is not possible to compute the prior probability, $P(g_i)$, directly from the above equation, we need to approximate it. There are several methods to approximate the prior probability, including variational inference [34], Markov chain Monte Carlo [35] and expectation propagation [36]. In this paper, we use the Expectation-Maximization (EM) algorithm [37] to estimate $P(g_i)$. The EM algorithm is particularly effective when the likelihood is well defined, which in our case corresponds to the obfuscation matrix, M.

Let DB be a bag of perturbed location data from sampled users. The EM process for estimating $P(g_i)$ for all $g_i \in G$ from DB is as follows.

- Initialization: The parameter (i.e., prior probability) is initialized as follows:

$$P^{(0)}(g_1) = P^{(0)}(g_2) = \cdots = P^{(0)}(g_m) = \frac{1}{m} \tag{5}$$

- E-step: The posterior probability is calculated based on the current parameters as follows:

$$P(g_i|g'_j) = \frac{P^{(t)}(g_i)M[i, j]}{\sum_{g_k \in G} P^{(t)}(g_k)M[k, j]} \tag{6}$$

- M-step: The parameter is updated using the current posterior probabilities calculated in the previous E-step:

$$P^{(t+1)}(g_i) = \frac{\sum_{g'_j \in DB} P(g_i|g'_j)}{|\mathcal{DB}|} \tag{7}$$

Here, $|DB|$ represents the number of data in DB. After updating the prior probabilities, we perform a normalization step to ensure that the sum of all prior probabilities equals 1 as follows:

$$P^{(t+1)}(g_i) = \frac{P^{(t+1)}(g_i)}{\sum_{g_k \in G} P^{(t+1)}(g_k)} \tag{8}$$

The above E-step and M-step are iterated until the parameter converges to a stable value or the number of iterations reaches a predefined threshold.

4.3. Computing Adaptively Partitioned Grids

In this subsection, we introduce a method that adaptively partitions grids based on the probability distribution (i.e., $P(g_i)$ for all $g_i \in G$) computed in the previous phase. Initially, the proposed method treats all grids in G as a single grid cluster, and then iteratively partitions this cluster in a top-down using a greedy algorithm.

Let $GC_v = \{C_1, C_2, \cdots, C_{|GC_v|}\}$ represent a set of grid clusters after the v-th partition. Assume that for each $C_k \in GC_v$, $grid(C_k) \subset G$ denotes the set of grids that belong to the cluster C_k. Let n be the total number of users from whom the server collects location data. Then, the expected number of users located in grid g_i is computed as $Cnt(g_i) = n \cdot P(g_i)$.

Furthermore, let M_{GC_v} be an $|GC_v| \times |GC_v|$ obfuscation matrix, satisfying ϵ-Geo-Ind, constructed over elements in GC_v using Equation (3). The distance between any two clusters necessary to compute M_{GC_v} is determined using the centroids of the grids belonging to each cluster. Then, assuming that users perturb their location according to the probabilities encoded in M_{GC_v}, the expected number of perturbed location data corresponding to grids belonging to C_K that the server receives from n users is computed as follows:

$$Cnt_{pert}(C_k) = \sum_{C_j \in GC_v} \sum_{g_i \in grid(C_j)} Cnt(g_i) \times M[j,k] \tag{9}$$

Let $Clus()$ be a function that takes a grid as input and outputs the cluster to which that grid belongs. Assuming that users are evenly distributed across the grids within each cluster, the expected error due to Geo-Ind with GC_v is computed as follows:

$$Err_{GC_v} = \sum_{g_i \in G} \left| Cnt(g_i) - \frac{Cnt_{pert}(Clus(g_i))}{size(Clus(g_i))} \right| \tag{10}$$

Here, $size(C_k)$ denotes the number of grids that belong to the cluster C_k.

Let us assume that in the next $(v+1)$-th partition, $C_h \in GC_v$ is selected to be divided into subclusters. In this paper, we partition C_h into four equal-sized subclusters by dividing the associated region horizontally and vertically. Let GC^h_{v+1} represent the set of grid clusters newly obtained by subdividing C_h. Using the method described above, we can similarly estimate the expected error, $Err_{GC^h_{v+1}}$, caused by Geo-Ind with GC^h_{v+1}. The set of grid clusters for the $(v+1)$-th iteration is then determined as follows:

$$GC_{v+1} = \underset{1 \le h \le |GC_v|}{\operatorname{argmax}} \left(Err_{GC^h_{v+1}} - Err_{GC_v} \right) \tag{11}$$

In other words, a grid cluster C_h that provides the maximum error reduction gain is selected for partitioning in the next iteration.

Algorithm 1 presents pseudocode for adaptively partitioning grids using the probability distribution of users. The algorithm takes as input a set of grids, G, and probability distributions, $P(g_1), \cdots, P(g_m)$, and outputs a set of grid clusters, GC. In line 1, GC_{cur} is initialized to contain a single cluster that includes all grids in G. Then, in line 3, $Err_{GC_{cur}}$ is computed using GC_{cur}. Between lines 4 and 12, the algorithm identifies the cluster $C_h \in GC_{cur}$ that yields the maximum error reduction gain. This process is repeated until the error reduction gain is greater than 0. Finally, the algorithm returns GC_{cur}.

Algorithm 1: Pseudo-code for adaptive grid partition

input : $G = \{g_1, \cdots, g_m\}$ and $P(g_i)$ for all $g_i \in G$
output: a set of grid clusters GC

Initialize GC_{cur};
while *true* **do**
  $Err_{GC_{cur}} = EstimateErr(GC_{cur}, M_{GC_{cur}})$;
  $Idx = 0, \quad Gain_{best} = -\infty$;
  for $h = 1$ **to** $|GC_{cur}|$ **do**
    $GC^h_{cur} = PartitionGrid(GC_{cur}, h)$;
    $Err_{GC^h_{cur}} = EstimateErr(GC^h_{cur}, M_{GC^h_{cur}})$;
    if $(Err_{GC^h_{cur}} - Err_{GC_{cur}}) > Gain_{best}$ **then**
      $Idx = h$;
      $Gain_{best} = Err_{GC^h_{cur}} - Err_{GC_{cur}}$;
    end
  end
  if $Gain_{best} > 0$ **then**
    $GC_{cur} = GC^{Idx}_{cur}$;
  else
    break
  end
end
return GC_{cur};

The proposed adaptive grid partitioning method in this subsection relies on the probability distribution of users estimated from sampled data. Thus, as with other sampling-based methods, there is a possibility that the sampled data may be biased. Such biases can result in a non-representative user distribution being used for grid partitioning, which can lead to ineffective partitioning. This, in turn, can adversely affect the overall utility of the collected location data, because the adaptive grids may not accurately reflect the true user density. In order to mitigate potential biases and inaccuracies in capturing real-time user distribution, spatial variations can be considered in the sampling process. One effective method is to use stratified sampling [38], which involves dividing the entire region into disjoint subregions. By ensuring that each subregion is proportionally represented in the sample, stratified sampling helps reduce sampling bias and provides a more accurate estimate of the user distribution.

4.4. Collecting Perturbed Location Data from Remaining Users Using Adaptively Partitioned Grids

A new obfuscation matrix, M, is computed using the adaptively partitioned grid set, GC, computed in the previous phase. The adaptively partitioned grids allow for a more precise and relevant obfuscation process by capturing the dynamic nature of the user distribution more effectively than static grids. Once the new obfuscation matrix is computed, it is distributed to the remaining users. These users then use the updated matrix to perturb their true location data, ensuring that their privacy is preserved according to the principles of Geo-Ind. The obfuscated location data are then sent back to the server, where it is integrated with the previously collected data from the sampled users.

5. Experiments

In this section, we first describe the experimental setup. Then, we discuss the experimental results.

5.1. Experimental Setup

In this section, we describe the experiments we carried out to evaluate the proposed approach. For our experiments, we used the Porto taxi trajectories dataset [39], which

consists of taxi trajectories composed of a series of GPS coordinates recorded from 442 taxis operating in the city of Porto, Portugal. We randomly extracted 50,000 location data from these trajectories, of which 10,000 were considered as location data of the sampled users. In the experiment, we varied the number of grids from 400 (i.e., 20-by-20 grids) to 10,000 (i.e., 100-by-100 grids). In the experiments, results are reported for the following alternatives: the existing non-adaptive grid (NG) method in [13], and the adaptive grid (AG) method introduced in this paper. We use the following metrics for evaluation:

- Data-level metric measures the similarity between the true location dataset and the perturbed location dataset collected under ϵ-Geo-Ind. For the data-level evaluation, we use both the average count error (ACE) and the density error. The average count error quantifies the difference between the actual number of users, $num_{true}(g_i)$, and the number derived from the perturbed dataset, $num_{pert}(g_i)$, for each grid. It is calculated as

$$Average\ count\ error = \sum_{1 \le i \le m} \frac{|num_{true}(g_i) - num_{pert}(g_i)|}{max(num_{true}(g_i), 1)} \tag{12}$$

  The density error measures the difference between the actual density distribution of users and the perturbed version computed from the datasets collected under ϵ-Geo-Ind. This error is measured as

$$Density\ error = JSD(D(OD), D(PD)) \tag{13}$$

  Here, $JSD()$ represents the Jenson–Shannon divergence between two distributions from the original location dataset, $D(OD)$ and from the perturbed location dataset, $D(PD)$.
- Application-level metric evaluates the utility of collected data from the perspective of applications that use it. We use range query error for this metric, a widely recognized measure for evaluating the effectiveness of location data [14]. In the experiment, we generate a range query, Q_R, with a random region R, and compare the number of results from the original location dataset, $Q_R(OD)$, with those from the perturbed location datasets, $Q_R(PD)$. It is calculated as

$$Range\ query\ error = \frac{|Q_R(OD) - Q_R(PD)|}{max(Q_R(OD), 1)} \tag{14}$$

  In the experiments, we generated 200 range queries and reported the average range query error.

In the experiment, various privacy budgets (ϵ) ranging from 0.2 to 2.0 were used. A privacy budget of less than 2 is typically considered acceptable in practical applications [14]. We implemented both NG and AG using Python 3.8, and all experiments were conducted in an environment equipped with Intel Xeon 5220R CPUs and 64 MB of memory.

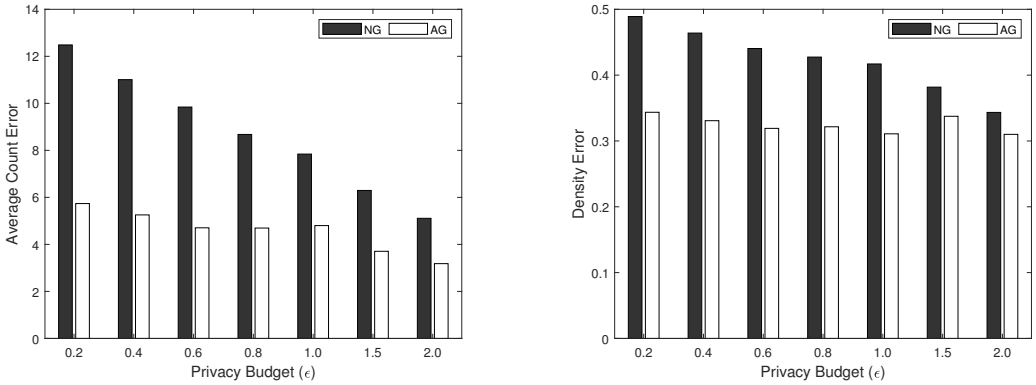
5.2. Results

In this subsection, we first present the results of evaluating the data level, and then present the results of evaluating the application level.

5.2.1. Data-Level Evaluation

Figure 2 shows the effect of the privacy budget on both the average count error and the density error. In this experiment, the privacy budget varies from 0.2 to 2.0, while the grid size is fixed at 400. As ϵ decreases, both errors increase. This is because as ϵ decreases, the degree of perturbation caused by Geo-Ind increases, leading to an increased error, which is commonly observed with DP-based methods. As shown in the figures, the proposed method (AG) consistently outperforms the existing method (NG) across all privacy budget levels. Furthermore, the performance gap between the two methods increases as the privacy

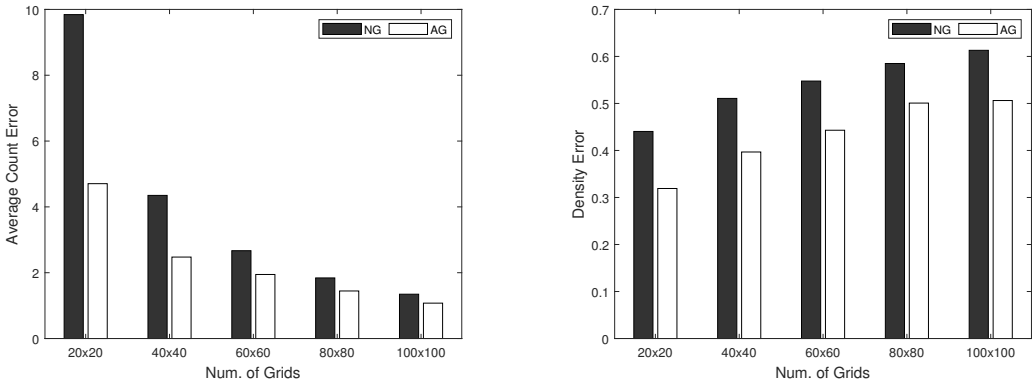
budget decreases, and thus the level of privacy increases. This shows that the proposed method is more advantageous for applications that require a high level of privacy, which is typical for most applications that handle location data.



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

Figure 2. Effect of the privacy budget on (**a**) the average count error and (**b**) the density error for the existing non-adaptive grid (NG) method and the proposed adaptive grid (AG) method.

Figure 3 shows the effect of the number of grids on the average count error and the density error. In this experiment, the number of grids varies from 400 to 10,000, while the privacy budget is fixed at 0.6. The figure shows that the proposed method consistently outperforms the existing method across all grid sizes. More specifically, as can be seen in the figure, the average count error decreases as the number of grids increases. Note that as the number of grids increases, the number of users per grid decreases because the total number of users is fixed. This, in turn, reduces the average count error, which is based on the absolute difference between the users obtained from the actual location data and the perturbed location data. On the other hand, as the number of grids increases, the density error, which measures the Jenson–Shannon divergence between two distributions from the original location data set and the perturbed location data sets, increases. This is because the Jenson–Shannon divergence measures the relative difference between distributions, and therefore, is not affected by the number of users per grid. As the grid size becomes finer, the perturbations in the data have a more pronounced effect on the distribution, resulting in a higher divergence between the original and perturbed datasets.



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

Figure 3. Effect of the number of grids on (**a**) the average count error and (**b**) the density error for the existing non-adaptive grid (NG) method and the proposed adaptive grid (AG) method.

The results shown in Figures 2 and 3 confirm that the proposed method enables the collection of location data that are more similar to the original data under Geo-Ind than the existing method. These results highlight the significant advantages of our approach

in privacy-preserving location data collection. By dynamically adjusting grids based on real-time user distribution under Geo-Ind, we achieve higher data accuracy and utility.

5.2.2. Application-Level Evaluation

Figure 4 illustrates the effect of the privacy budget and the number of grids on the range query error. In Figure 4a, the grid size is fixed at 400, while in Figure 4b, the privacy budget is fixed at 0.6. As shown in the figure, as ϵ decreases, the error associated with the existing method increases dramatically, while the error for the proposed method increases only marginally. This occurs because the proposed method is able to collect location datasets that are closer to the original datasets, as verified by the data-level evaluation. The robustness of our approach under varying privacy budgets highlights its effectiveness in balancing privacy and accuracy. As ϵ becomes smaller, indicating stronger privacy guarantees, the proposed method still manages to preserve the utility of the data, making them more reliable for applications that require precise location information.

Moreover, the proposed method consistently outperforms the existing method across all grid sizes. This verifies that our proposed method is robust regardless of the number of grids. The ability to maintain low query error rates across different grid configurations demonstrates the adaptability and effectiveness of our approach. This robustness is crucial for practical applications that require different granularity in representing user location (i.e., number of grids) depending on the application requirements.

These experimental results indicate that the proposed method can be used for a wide range of location-based services and applications requiring different privacy levels and granularity in representing user location.

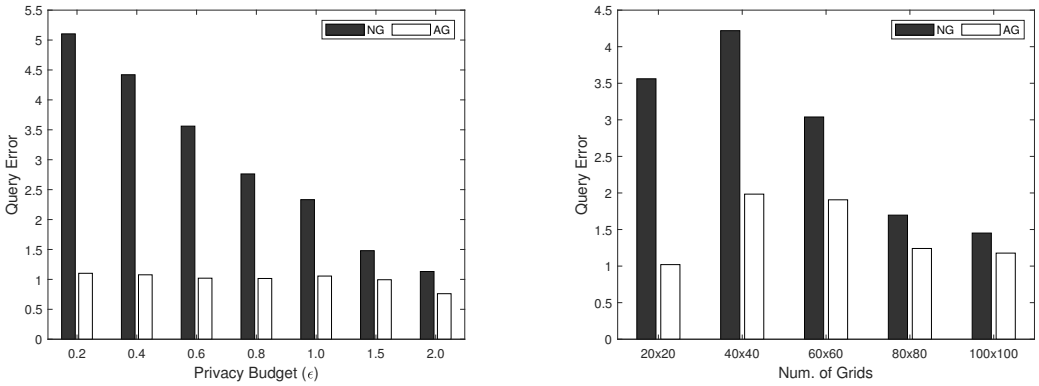


Figure 4. Effect of (**a**) the privacy budget and (**b**) the number of grids on the range query error for the existing non-adaptive grid (NG) method and the proposed adaptive grid (AG) method.

5.2.3. Evaluation of Network Variability and Grid Adaptation Effects

All experiment results in the previous subsections were obtained under the assumption that the network conditions are stable. However, in real-world scenarios, network conditions can be highly variable and unpredictable. The proposed method adaptively partitions the grid based on the sampled location data collected during the process of location data collection. However, unstable network conditions, such as high network latency, packet loss, and low bandwidth, can delay the timely collection of these sampled data. As a result, some sampled data may not be available for the computation of adaptive grid partitioning, which may lead to less accurate grid partitioning.

In this subsection, to address the challenges of unstable networks, we evaluate the effectiveness of the proposed method in real network scenarios. The experiment shown in Table 1 considers a scenario where some of the sampled location data needed to estimate the user distribution in Section 4.2 (which is then used to adaptively compute the grid partitions in Section 4.3) is either lost or not received in time due to unstable network

conditions such as high latency, packet loss, and low bandwidth. In this experiment, the loss rate of sampled location data varies from 1% to 20%, covering a range from typical to severe network conditions. In the experiment, the number of grids is set to 400 and the privacy budget is set to 0.6. As shown in Table 1, even as the loss rate of sampled location data increases from 1% to 20%, both errors remain stable with only a very small increase. In particular, even under severe network conditions with a 20% loss rate, the proposed adaptive grid (AG) method significantly outperforms the non-adaptive grid (NG) method. These results verify the robustness and effectiveness of the proposed method in maintaining data utility under real network scenarios with varying levels of data loss due to unstable network conditions.

Table 1. Effect of the loss rate of sampled location data on the average count error and the density error.

	AG					NG
Loss Rate of Sampled Location Data	0%	1%	5%	10%	20%	
Average Count Error	4.705	4.867	4.896	4.936	5.018	9.842
Density Error	0.319	0.321	0.328	0.336	0.341	0.440

The proposed method adaptively adjusts the grid during the location data collection process, which introduces additional latency due to the computational overhead of computing the adaptive grid. Hence, we experimentally evaluate the latency introduced by the proposed method. Table 2 shows the latency results caused by the adaptive grid computation of the proposed method. In this experiment, the number of grids varies from 400 to 10,000, while the privacy budget is fixed at 0.6. As shown in the table, the latency increases as the grid size increases. This is because as the number of grids increases, the number of iterations required to adaptively partition the grids also increases. As a result, larger grids require more computational resources, which leads to higher latency.

Table 2. Latency caused by computing the adaptive grid.

Grid size	20×20	40×40	60×60	80×80	100×100
Execution time (s)	4.10	17.41	43.12	139.56	439.56

We note that although the computation of the adaptive grid introduces additional latency as shown in Table 2, it is a one-time process within the overall location data collection procedure. Therefore, the impact of this overhead on the overall processing time of the location data collection is limited. Furthermore, the additional latency caused by the computational overhead of adaptively partitioning the grid can be mitigated by various parallel processing techniques. In particular, techniques such as distributed computing frameworks [40,41] and GPU acceleration can significantly reduce computation time, thereby mitigating this latency. By distributing the workload across multiple processors, these approaches can improve the efficiency of the adaptive grid partitioning process, ensuring data analysis in real-time applications.

6. Conclusions and Future Work

Recently, there has been an increasing demand for the collection and sharing of location data. Given the sensitive nature of user location information, considerable efforts have been made to ensure privacy, with differential privacy-based schemes emerging as the preferred approach. However, these schemes typically represent user locations on uniformly partitioned grids, which often do not accurately reflect the true distribution of users within a space. In this paper, we presented a novel approach that dynamically adjusts the grid in real-time during location data collection using Geo-Ind to enhance the utility of the collected data. The proposed method captures the user distribution directly during data collection, eliminating the reliance on pre-existing distribution information. Experimental

results on real data confirmed that the proposed scheme significantly improves the utility of the collected location data at both the data and application levels. Specifically, the results showed that compared to the existing solution, the proposed method can reduce the error rate by up to 52% in data-level experiments and by up to 75% in application-level experiments.

Despite the promising results, the proposed method has the following limitations. Since the proposed method adaptively adjusts the grid in real-time during data collection, there is an additional computational overhead associated with computing the adaptive grid. This overhead is especially significant when using large grid sizes. Thus, future work will focus on improving the efficiency of the adaptive grid computation, especially for large grid sizes with large numbers of users. This can be achieved by parallelizing the adaptive grid partitioning process to reduce the computational overhead. We will explore various parallel processing techniques, such as implementing distributed computing frameworks such as Apache Hadoop [40] or Apache Spark [41], which distribute data and computation across a cluster of machines. In addition, multi-threading within a single machine and the use of GPU acceleration can be considered to increase efficiency. Furthermore, integrating cloud computing services will improve scalability by providing a dynamic and scalable infrastructure for performing adaptive grid partitioning on large datasets.

Another future research direction is to theoretically analyze the impact of adaptive grid partitioning on the data utility of collected location data. This analysis will elucidate the underlying principles of adaptive grid partitioning and its impact on data utility, thus providing more robust theoretical support for the proposed method. In addition, the privacy-utility tradeoff can be further investigated to optimize the balance between privacy and utility. This will include the development of models and metrics to quantitatively assess this trade-off under various conditions.

Funding: This research was funded by a 2023 Research Grant from Sangmyung University (2023-A000-0119).

Data Availability Statement: The original data presented in the study are openly available in Kaggle at https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i (accessed on 20 June 2024).

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DP	Differential Privacy
LDP	Local Differential Privacy
MDP	Metric Differential Privacy
Geo-Ind	Geo-Indistinguishability
MCS	Mobile Crowdsensing
EM	Expectation-Maximization
JSD	Jenson–Shannon Divergence

References

1. Wang, X.; Ma, Y.; Wang, Y.; Jin, W.; Wang, X.; Tang, J.; Jia, C.; Yu, J. Traffic flow prediction via spatial temporal graph neural network. In Proceedings of the Web Conference, Taipei, Taiwan, 20–24 April 2020; pp. 1082–1092.
2. Pan, Z.; Liang, Y.; Wang, W.; Yu, Y.; Zheng, Y.; Zhang, J. Urban traffic prediction from spatio-temporal data using deep meta learning. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 April 2019; pp. 1720–1730.
3. Kim, J.S.; Kim, J.W.; Chung, Y.D. Successive Point-of-Interest Recommendation With Local Differential Privacy. *IEEE Access* **2021**, *9*, 66371–66386. [CrossRef]
4. An, J.; Li, G.; Jiang, W. NRDL: Decentralized user preference learning for privacy-preserving next POI recommendation. *Expert Syst. Appl.* **2024**, *239*, 122421. [CrossRef]
5. Primault, V.; Boutet, A.; Mokhtar, S.B.; Brunie, L. The long road to computational location privacy: A survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2772–2793. [CrossRef]

6. Liu, B.; Zhou, W.; Zhu, T.; Gao, L.; Xiang, Y. Location privacy and its applications: A systematic study. *IEEE Access* **2019**, *6*, 17606–17624. [CrossRef]
7. Alharthi, R.; Banihani, A.; Alzahrani, A.; Alshehri, A.; Alshahrani, H.; Fu, H.; Liu, A.; Zhu, Y. Location privacy challenges in spatial crowdsourcing. In Proceedings of the IEEE International Conference on Electro/Information Technology, Rochester, MI, USA, 3–5 May 2018.
8. Henriksen-Bulmer, J.; Jeary, S. Re-identification attacks—A systematic literature review. *Int. J. Inf. Manag.* **2016**, *36*, 1184–1192. [CrossRef]
9. Dwork, C. Differential privacy. In Proceedings of the International Colloquium on Automata, Languages, and Programming, Venice, Italy, 10–14 July 2006; pp. 1–12.
10. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating noise to sensitivity in private data analysis. In Proceedings of the Third conference on Theory of Cryptography, New York, NY, USA, 4–7 March 2006.
11. Bordenabe, N.E.; Chatzikokolakis, K.; Palamidess, C. Optimal geo-indistinguishable mechanisms for location privacy. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 3–7 November 2014; pp. 251–262.
12. Kim, J.; Jang, B. Workload-aware indoor positioning data collection via local differential privacy. *IEEE Commun. Lett.* **2019**, *23*, 1352–1356. [CrossRef]
13. Zhang, P.; Cheng, X.; Su, S.; Wang, N. Area coverage-based worker recruitment under geo-indistinguishability. *Comput. Netw.* **2022**, *217*, 109340. [CrossRef]
14. Du, Y.; Hu, Y.; Zhang, Z.; Fang, Z.; Chen, L.; Zheng, B.; Gao, Y. LDPTrace: Locally differentially private trajectory synthesis. In Proceedings of the VLDB Endowment, Vancouver, BC, Canada, 28 August–1 September 2023; pp. 1897–1909.
15. Ghaemi, Z.; Farnaghi, M. A Varied Density-based Clustering Approach for Event Detection from Heterogeneous Twitter Data. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 82. [CrossRef]
16. Alvim, M.; Chatzikokolakis, K.; Palamidessi, C.; Pazii, A. Local Differential Privacy on Metric Spaces: Optimizing the Trade-Off with Utility. In Proceedings of the IEEE Computer Security Foundations Symposium, Oxford, UK, 9–12 July 2018.
17. Andres, M.E.; Bordenabe, N.E.; Chatzikokolakis, K.; Palamidessi, C. Geo-indistinguishability: Differential privacy for location-based systems. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Berlin, Germany, 4–8 November 2013; pp. 901–914.
18. Chatzikokolakis, K.; Palamidessi, C.; Stronati, M. Geo-indistinguishability: A principled approach to location privacy. In Proceedings of the International Conference on Distributed Computing and Internet Technology, Bhubaneswar, India, 5–8 February 2015; pp. 49–72.
19. Wang, L.; Yang, D.; Han, X.; Wang, T.; Zhang, D.; Ma, X. Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation. In Proceedings of the International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 627–636.
20. Qiu, C.; Squicciarini, A.C. Location privacy protection in vehicle-based spatial crowdsourcing via geo-indistinguishability. In Proceedings of the IEEE International Conference on Distributed Computing Systems, Dallas, TX, USA, 7–10 July 2019; pp. 1061–1071.
21. Jin, W.; Xiao, M.; Guo, L.; Yang, L.; Li, M. ULPT: A user-centric location privacy trading framework for mobile crowd sensing. *IEEE Trans. Mob. Comput.* **2022**, *21*, 3789–3806. [CrossRef]
22. Huang, P.; Zhang, X.; Guo, L.; Li, M. Incentivizing crowdsensing-based noise monitoring with differentially-private locations. *IEEE Trans. Mob. Comput.* **2021**, *20*, 519–532. [CrossRef]
23. Zhao, Y.; Yuan, D.; Du, J.T.; Chen, J. Geo-Ellipse-Indistinguishability: Community-aware location privacy protection for directional distribution. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 6957–6967. [CrossRef]
24. Yu, L.; Zhang, S.; Meng, Y.; Du, S.; Chen, Y.; Ren, Y.; Zhu, H. Privacy-preserving location-based advertising via longitudinal geo-indistinguishability. *IEEE Trans. Mob. Comput.* **2024**, *23*, 8256–8273. [CrossRef]
25. Zhao, Y.; Chen, J. Vector-indistinguishability: Location dependency based privacy protection for successive location data. *IEEE Trans. Comput.* **2024**, *73*, 970–979. [CrossRef]
26. Mendes, R.; Cunha, M.; Vilela, J.P. Velocity-aware geo-indistinguishability. In Proceedings of the ACM Conference on Data and Application Security and Privacy, Charlotte, NC, USA, 24–26 April 2023; pp. 141–152.
27. Ren, W.; Tang, S. EGeoIndis: An effective and efficient location privacy protection framework in traffic density detection. *Veh. Commun.* **2020**, *21*, 100187. [CrossRef]
28. Kim, J.W.; Lim, B. Effective and Privacy-Preserving Estimation of the Density Distribution of LBS Users under Geo-Indistinguishability. *Electronics* **2023**, *12*, 917. [CrossRef]
29. Chen, R.; Li, L.; Ma, Y.; Gong, Y.; Guo, Y.; Ohtsuki, T.; Pan, M. Constructing Mobile Crowdsourced COVID-19 Vulnerability Map With Geo-Indistinguishability. *IEEE Internet Things J.* **2022**, *9*, 17403–17416. [CrossRef]
30. Fathalizadeh, A.; Moghtadaiee, V.; Alishahi, M. Indoor Geo-Indistinguishability: Adopting Differential Privacy for Indoor Location Data Protection. *IEEE Trans. Emerg. Top. Comput.* **2023**, *12*, 293–306. [CrossRef]
31. Feyisetan, O.; Balle, B.; Drake, T.; Diethe, T. Privacy-and utility-preserving textual analysis via calibrated multivariate perturbations. In Proceedings of the International Conference on Web Search and Data Mining, Houston, TX, USA, 3–7 February 2020; pp. 178–186.

32. Song, S.; Kim, J.W. Adapting Geo-Indistinguishability for Privacy-Preserving Collection of Medical Microdata. *Electronics* **2023**, *12*, 2793. [CrossRef]
33. Ahuja, R.; Ghinita, G.; Shahabi, C. A utility-preserving and scalable technique for protecting location data with geo-indistinguishability. In Proceedings of the International Conference on Extending Database Technology, Lisbon, Portuga, 26–29 March 2019; pp. 210–231.
34. Blei, D.M.; Kucukelbir, A.; McAuliffe, J.D. Variational Inference: A Review for Statisticians. *J. Am. Stat. Assoc.* **2017**, *112*, 859–877. [CrossRef]
35. Chib, S. Markov Chain Monte Carlo Methods: Computation and Inference. *Handb. Econom.* **2001**, *5*, 3569–3649.
36. Li, Y.; Hernandez-Lobato, J.M.; Turner, R.E. Stochastic Expectation Propagation. *arXiv* **2018**, arXiv:1506.04132.
37. Sammaknejad, N.; Zhao, Y.; Huang, B. A review of the Expectation Maximization algorithm in data-driven process identification. *J. Process Control* **2019**, *73*, 123–136. [CrossRef]
38. Howell, C.R.; Su, W.; Nassel, A.F.; Agne, A.A.; Cherrington, A.L. Area based stratified random sampling using geospatial technology in a community-based survey. *BMC Public Health* **2020**, *20*, 1678. [CrossRef] [PubMed]
39. Moreira-Matias, L.; Gama, J.; Ferreira, M.; Mendes-Moreira, J.; Damas, L. Predicting taxi–passenger demand using streaming data. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1393–1402. [CrossRef]
40. Apache Hadoop. Available online: https://hadoop.apache.org/ (accessed on 22 July 2024).
41. Apache Spark—Unified Engine for Large-Scale Data Analytics. Available online: https://spark.apache.org/ (accessed on 22 July 2024).