Publisher: State and Provincial Joint Engineering Lab. of Advanced Network

Monitoring and Control (ANMC)

Cooperate:

Xi'an Technological University (CHINA)
West Virginia University (USA)
Huddersfield University of UK (UK)
Missouri Western State University (USA)
James Cook University of Australia
National University of Singapore (Singapore)

Approval:

Library of Congress of the United States

Shaanxi provincial Bureau of press, Publication, Radio and Television

Address:

4525 Downs Drive, St. Joseph, MO64507, USA

No. 2 XueFu Road, WeiYang District, Xi’an, 710021, China

Telephone: +1-816-2715618 (USA) +86-29-86173290 (CHINA)

Website: www.ijanmc.org

E-mail: ijanmc@ijanmc.org

xxwlcn@163.com

ISSN: 2470-8038

Print No. (China): 61-94101

Publication Date: March 23, 2025

Would you like to be an Associate Editor? Simply send a request together with your Curriculum Vitae to xxwlcn@163.com. We will have a team of existing editors or at least three experts in your field to review your request and make a decision as soon as we can. The criteria to be an associate editor are: 1. must have advanced degree; 2. must be a leader or have outstanding achievements in the specific research field; 3. must be recommended by the review team.

Table of Contents

Improvement of Remote Sensing Target Tracking Method Based on Deep Learnin

Xuhao Wang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: sususama2@163.com

Long Ma
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: malong@xatu.edu.cn

***Abstract*—Remote Sensing Object Tracking refers to the process of detecting, recognizing, and tracking targets on the ground or at sea using remote sensing technology, particularly sensors mounted on satellite or aerial platforms to obtain high-resolution remote sensing image sequences. Current methods for remote sensing object tracking face challenges such as low tracking success rates and inefficiencies. This paper proposes a neural network for remote sensing object tracking based on SiamRPN++, which introduces an improved network structure incorporating the C3Minus module and a coordinate attention mechanism within the backbone extraction network. Furthermore, we design a feature extraction module, ResSwinT, that combines ResNet and Swin Transformer architectures to integrate local and global information obtained from feature maps as foundational features. This approach effectively addresses the aforementioned issues, and quantitative experiments demonstrate an increase in accuracy and success rates by 1.9% and 4.7%, respectively, indicating that our method effectively handles object tracking in remote sensing images.**

Keywords-Deep Learning; Object Tracking; Remote Sense

I. INTRODUCTION

In this paper, remote sensing targets refer to various objects captured in remote sensing images by satellite sensors in the visible light spectrum, specifically within the range of 0.38 to 0.76 micrometers. These images contain a wealth of detailed information, effectively reflecting the shape, color, and texture of terrestrial objects, making them easily observable by the human eye. Target recognition represents a significant area of research within the field of computer vision, with the objective of detecting and tracking objects of interest within video sequences.

The core task of target tracking involves predicting a target's future position and dimensions, starting from its initial state in the first video frame. This task comprises several crucial components: motion models, feature extraction, observation models, model updating, and ensemble methods. The motion model generates samples, feature extraction represents the target, the observation model evaluates the samples, model updating adjusts for target changes, and ensemble methods combine decisions for improved predictions.

The field of remote sensing would be incomplete without the technology of remote sensing object tracking, which plays a pivotal role in the discipline. It involves detecting and identifying specific targets or features from remote sensing images, where the targets typically refer to roads, buildings, vehicles, aircraft, and ships. With advancements in remote sensing technology, remote sensing images now possess broad perspectives and ultra-high spatial and temporal resolutions, and they are minimally affected by variations in viewing angles and lighting conditions. The development of high resolution remote sensing portraits had increasingly emphasized the importance of remote sensing object recognition in various fields such as urban planning, environmental monitoring, agriculture, military applications, and disaster management. However, this advancement has also introduced new challenges within the remote sensing domain. As illustrated in Figure 1, remote sensing object difficulty recognition significantly increases in

Figure 1. Shows remote sensing portraits. The left portrait is the original image, which contains a lot of information. The right image is a partial image captured from the left image. It can be seen that even if it is magnified many times, the information in the image is still very rich

high-resolution images due to complex backgrounds and the small size of targets relative to the overall image.

To address the aforementioned issues, this study employs deep learning techniques to improve the SiamRPN++ [1] algorithm, enhancing its resolution and feature extraction capabilities in complex backgrounds, thereby significantly improving the tracking performance of small targets in remote sensing portraits. In modified neural network architecture, we introduce the C3Minus module structure and coordinate attention mechanism [2] to optimize the feature extraction capabilities of the backbone network. Furthermore, this study design a composite feature extraction module, ResSwinT, which combines CNN [3] and Swin Transformer [4]. This module effectively integrates local and global feature information, providing a richer representation foundation for the underlying features.

II. Related Works

Remote sensing target tracking can be classified into classical tracking methods, correlation filter-based tracking methods, and deep learning-based tracking methods.

Classical tracking methods mainly include Kalman filtering, particle filtering, and Bayesian estimation. Kalman filtering provides optimal estimates only within Gaussian linear models. To address this limitation, Kulikova [5] proposed a NIRK-based precise continuous-discrete extended Kalman filter, while Zhou Huan et al. [6] introduced an adaptive unscented Kalman filter for target tracking in nonlinear systems involving model mismatches, which can handle divergence caused by sensor failures or model mismatches during the tracking process. Particle filtering is a Monte Carlo-based method that models the target state as a set of particles, where each particle represents a possible estimate of the target state. The particles propagate over time according to a process model and update their weights based on measurement information. Particle filtering has been applied to target tracking in remote sensing images by modeling the target state as a set of variables and updating the particle weights according to remote sensing data. Bayesian estimation is a probabilistic method for target tracking that models the target state as a probability distribution. It updates the distribution based on measurement information and former knowledge of the target condition. Bayesian calculation represents the target state as a set of variables and updates the distribution according to remote sensing data. The Smooth Variable Structure Filter (SVSF) [7] was the model-based estimation method suitable for smoothing nonlinear dynamic systems. It accounts for sources of uncertainty and ensures stability the maximum limits on uncertainty and noise levels, with performance improvements achievable through finer definitions of parameter variations or uncertainty bounds.

Methods based on correlation filtering begin with initializing a target template, which is defined according to the initial target position. This template typically encompasses the target and surrounding pixels within a region of interest (ROI), and it is updated throughout subsequent video frames. S. Xuan et al. [8] developed a correlation filter based on embedded motion estimation to address the tracking of rapidly moving targets in satellite videos. Y. Liu [9] proposed a novel method employing an embedded multi-feature fusion and motion trajectory compensation kernelized correlation filter for tracking fast-moving targets in satellite videos. This approach utilizes a multi-feature fusion strategy to comprehensively describe the target, thus addressing issues of tracking accuracy caused by inaccurate target localization. Additionally, it incorporates adaptive Kalman filtering to compensate for the kernel correction in the KCF tracker, reducing boundary box drift of the objects.

In recent years, with the development of neural networks, an increasing number of remote sensing image target tracking methods have employed convolutional neural networks (CNNs), recurrent neural networks (RNNs), and attention mechanisms. Some studies utilize pre-trained CNNs, such as VGG [10] and ResNet [11], to extract features from remote sensing images, while others focus on training CNNs from scratch using large datasets. Target tracking networks based on Siamese networks have also been applied by Bertinetto et al. [12] They introduced SiamFC, which integrates Siamese networks into target tracking. However, SiamFC relies on exhaustive multi-scale search to regress the target bounding box, resulting in low efficiency and accuracy. Based on SiamFC, Li et al.[13] proposed SiamRPN, which integrates previous correlation filtering methods. It uses the first frame as a detection template while improving the output of SiamFC by enhancing the twin network's output layer. This enhancement allows for the extraction of proposed regions and scoring, resulting in more accurate target bounding boxes. Zhu et al.[14] introduced DaSiamRPN, which focuses more on semantic interference factors and incorporates an interference-aware module for incremental learning. Building on SiamRPN, Li et al.[1] developed SiamRPN++, which employs multi-layer aggregation to fuse shallow and deep features during feature extraction. This approach leverages modern deep neural networks' capability to capture features, significantly enhancing the model's accuracy. Wang et al. [15] presented SiamMask, which combines segmentation concepts with the twin network, adding a mask branch during the regression phase to compute the loss of the segmentation network. Xu et al. [16] proposed SiamFC++, which employs an anchor-free design, allowing the network to directly classify and regress candidate boxes for each position of the corresponding features, thus avoiding the impact of predefined anchors on network speed. In the field of remote sensing object tracking, Yan et al. [17] proposed a novel search algorithm, LightTrack, which encodes all possible frames into a BackBone supernet and a head supernet, significantly reducing inference time and thus speeding up the overall tracking process. Cao et al. [18] introduced online temporal adaptive convolution and an adaptive temporal transformer. The former dynamically adjusts convolutional weights using temporal information from previous frames to enhance spatial features, while the latter employs efficient temporal encoding to adjust similarity maps accurately, thereby improving the network's temporal awareness. Zhou et al. [19] integrated computer vision with natural language processing by unifying visual localization and object tracking into a single task. This framework leverages a multi-source relational module in the Transformer to effectively build a multimodal network structure. Hong et al. [20] developed a unified visual object tracking framework, OneTracker, which can handle multiple tracking tasks, including conventional RGB and RGB+X tracking, where X represents additional information, such as natural language descriptions, depth, thermal imaging, or event maps. OneTracker abstracts the common characteristics of tracking tasks and extends based on them to adapt to various tracking scenarios.

III. METHODS

This study proposes an improved overall network structure based on SiamRPN++, aiming to enhance tracking accuracy in the context of remote sensing images. The architecture of the entire network is illustrated in the figure below.

The network architecture employed in this research builds upon the foundational framework of SiamRPN++, incorporating significant modifications to several key modules. The architecture is systematically divided into three distinct components. The first component is the feature extraction module. In this study, we have modified the ResNet backbone, enhancing its enhancing feature extraction by incorporating the C3Minus and CA modules. These enhancements are designed to bolster the network's ability to extract relevant features from remote sensing images effectively. The second component consists of the ResSwinT module, which consolidates image patches from deeper layers to construct hierarchical feature maps. This approach significantly augments the network's modeling capacity, enabling it to capture complex representations within the input data. Finally, the third component is the regression head, which facilitates the precise localization of the target within the search space. This component is critical for ensuring that the network can accurately identify and track the target across varying conditions in remote sensing applications.

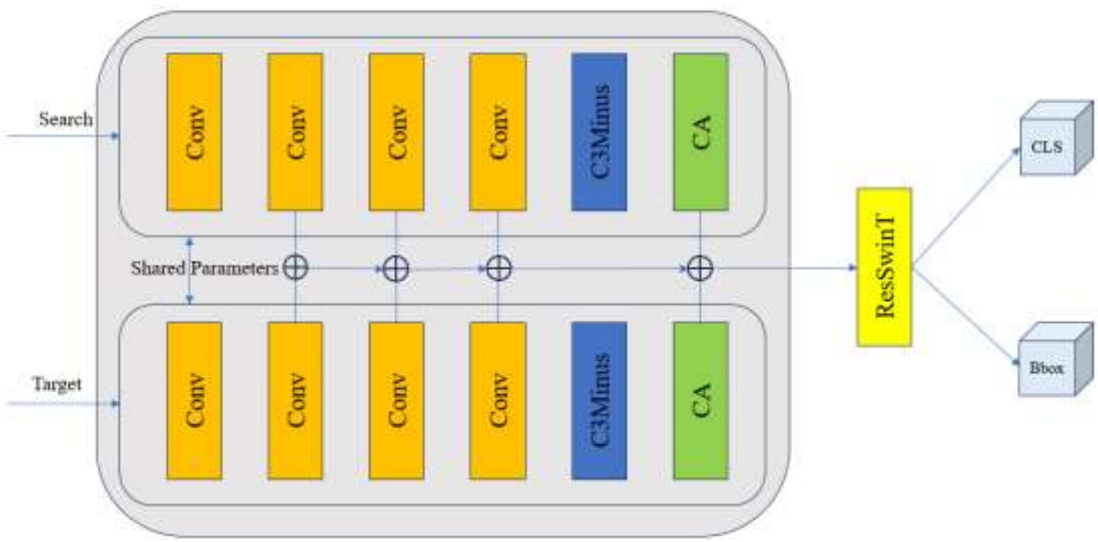


Figure 2. Network Structure. The proposed network's architecture is organized into three main components: the Backbone, the ResSwinT module, and the regression head. The Backbone is constructed upon the ResNet-50 architecture, with significant enhancements that include the incorporation of C3Minus and CA modules. These additions are designed to enhance the ability to extract features. The network effectively fuses the outputs from three distinct convolutional layers along with the outputs from the CA module. This fused information is subsequently passed to the ResSwinT module, which processes the data to generate hierarchical feature representations. Finally, the output from the ResSwinT module is directed to the regression head, which is responsible for accurately locating the target object in the image.

A. C3Minus

C3Minus module represents a significant advancement over the traditional CSPBottleneck block. It incorporates several key enhancements that contribute to improved performance and efficiency in convolutional neural networks. The specific improvements offered by C3Minus are as follows:

- **Efficiency**: C3Minus optimizes the convolutional operations by combining multiple individual convolutions into a single convolution layer. This strategic integration reduces the overall number of operations required during the forward pass, leading to reduced memory consumption and enhanced convolution efficiency. Consequently, this allows the network to process inputs more swiftly while maintaining performance.

- **Simplified Architecture**: By streamlining the structure of the network, C3Minus eliminates redundant convolutions that do not contribute significantly to feature extraction. This simplification results in a more compact architecture, which not only facilitates faster training times but also eases the deployment of the model in resource-constrained environments.

- **Reduced Computational Cost**: The C3Minus module effectively integrates convolutions, which leads to a decrease in both the number of parameters and the overall computational costs associated with the model. This reduction in complexity not only shortens the training duration but also enhances the model's scalability, allowing it to be more adaptable to various tasks and datasets.

The C3Minus module is of paramount importance in enhancing the performance of network by streamlining architecture, improving operational efficiency, and minimizing computational demands.

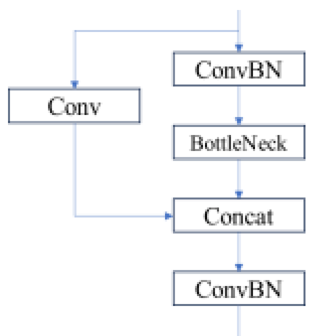


Figure 3. C3Minus network structure. The network consists of three convolution layers and one BottleNeck layer. ConvBN refers to the Batch Normalization and activation function, and Concat is a short circuit

B. CA

The CA module, which stands for Coordinate Attention, introduces a novel attention mechanism specifically designed for lightweight neural networks. Proposed by Hou et al.[2], this mechanism effectively integrates positional information into the channel attention framework, improving the model's capacity to detect critical spatial correlations and extended dependencies in the input data.

The CA module operates through two fundamental steps:

- **Coordinate Information Fusion**: In this initial phase, the module incorporates spatial coordinate information, which serves as a crucial component in understanding the spatial arrangement of features. By fusing this positional information with the channel-wise features, the CA module creates a more comprehensive representation that reflects the significance of different spatial locations relative to the features being processed. This fusion allows the network to prioritize certain features based on their spatial context, thereby improving its ability to discern important patterns.

- **Coordinate Attention Generation**: Following the fusion of coordinate information, the module generates coordinate attention maps. These maps are designed to emphasize relevant features across various spatial coordinates, effectively guiding the network to focus on critical areas within the input image. The generation of coordinate attention maps enables the network to adaptively weigh features based on their spatial context, enhancing the overall representation capabilities of the model.

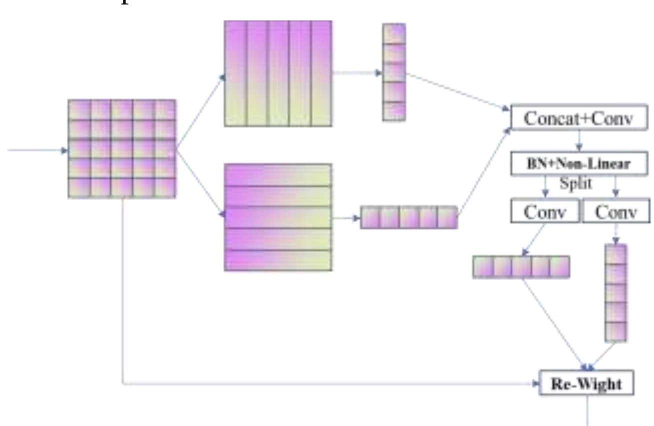


Figure 4. CA network structure. The entire module performs average pooling in the horizontal and vertical directions, then uses Transform to encode the spatial information, and finally fuses the spatial information by weighting it on the channel, making the network's overall perception of space more profound.

C. ResSwinT

In this study, the ResSwinT module is developed as an advanced extension of the ResNet block, integrating Swin Transformer layers to leverage the strengths of both convolutional neural networks (CNNs) and transformers. Inclusion of Swin Transformer layers allows for the construction of hierarchical feature representations, which are essential for capturing complex patterns and relationships within the data.

The ResSwinT module operates by utilizing the local and global attention mechanisms inherent in the Swin Transformer architecture. This enables the network to effectively learn contextual information across various scales, improving its capacity for discern intricate details in the input images. Specifically, hierarchical representations derived from the transformer layers facilitate the construction of detailed feature maps, which are crucial for accurately identifying and tracking targets.

One of the key advantages of the ResSwinT module is its capability to enhance local detail capture at lower stages of the network. By focusing on fine-grained features at these early stages, the module ensures that essential information is preserved and emphasized throughout the subsequent processing layers. This focus on local details is particularly beneficial in remote sensing applications, where variations in target appearance and background complexity can pose significant challenges.

Moreover, the integration of the Swin Transformer layers introduces a flexible windowing mechanism that enables the model to adaptively adjust its concentrate on different regions from the input image. This adaptability not only improves the model's efficiency but also enhances its performance in diverse scenarios, making it robust against various conditions encountered in real-world tracking tasks.

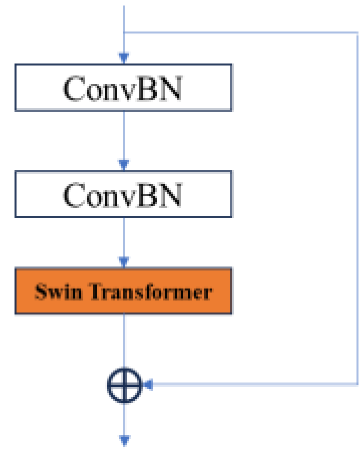


Figure 5. ResSwinT network structure, the overall structure uses a RestNet module as the basis, adds a Swin Transformer layer, and further extracts and fuses image features.

D. Detection Head

In this study, the detection head processes the outputs from the ResSwinT module to produce two distinct outputs: a binary classification output and a regression output. The main goal of binary classification is to distinguish the target object from the background within the search area. Meanwhile, the regression output is planned determine the accurate location of the target.

Regression head employs deep cross-correlation convolution to assess the relationship between the search area and the target template. This operation begins with feature maps secured since both the template and search branches, which were processed in batches to ensure they have the same number of channels. Subsequently, these two feature maps undergo a channel-wise correlation operation (essentially a convolution operation) to produce a result that maintains same number of channels. Finally, resulting feature maps are normalized to effectively merge the outputs from different channels. To finalize the process, an extra convolutional layer is included to produce the ultimate classification and regression outcomes.

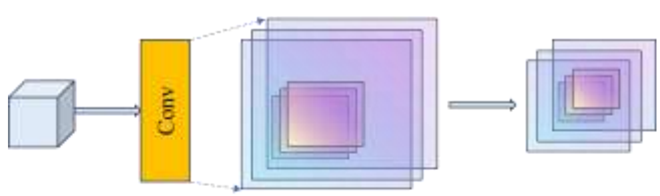


Figure 6. Depth-wise Cross Correlation

E. Loss Function and Optimizer

Since the network employed in this study outputs both classification and regression results, a mixed loss function[21] is used to measure the discrepancy between each branch output and the ground truth. The overall loss function is defined as:

$$Loss = \alpha L_{cls} + \beta L_{reg} \tag{1}$$

Where α and β are the weight coefficients for balancing the classification and regression loss components, set to 0.3 and 0.7 in this study, respectively. L_{cls} and L_{reg} represent the classification loss and regression loss functions. As the Backbone used in this study is based on ResNet-50, a model with strong classification capabilities, we assign a higher weight (0.7) to the classification component.

The classification output is responsible for distinguishing pixels within the search region as either background or target of interest, essentially a binary classification task. Therefore, we adopt the binary cross-entropy loss function:

$$L_{cls} = -y \times log(\hat{y}) - (1-y) \times log(1-\hat{y}) \tag{2}$$

where y denotes the ground truth, and $\hat{y}$ represents the predicted binary output from the network.

For the regression task, the output represents four absolute coordinates within the search region (upper left, upper right, lower left, and lower right), relative to the lower left corner of the region. Given that this task is essentially a regression problem, we apply the L_1 loss function[22], which is commonly used in regression tasks for its interpretability and tendency to produce sparse solutions:

$$L_1(x, \hat{x}) = \frac{1}{m} \sum_{i=1}^{m} \left| x_{(i)} - \hat{x}_{(i)} \right| \tag{3}$$

where x represents the ground truth, $\hat{x}$ is the network's regression output, and mmm is the number of coordinates in each sample.

The optimizer used in this study is Stochastic Gradient Descent (SGD). The SGD optimizer requires parameters such as the list of trainable parameters, momentum, and weight decay. It enhances training efficiency by accelerating the updates along relevant directions and reducing oscillations in irrelevant directions. The update process is given by:

$$V(t) = \gamma V(t-1) + \varepsilon \nabla Loss(\theta) \tag{4}$$

where t is the iteration count, ε is the learning rate, and $\nabla Loss(\theta)$ is the gradient of the loss with respect to the model parameters θ at the current iteration. Finally, parameters are updated as:

$$\theta = \theta - V(t) \tag{5}$$

Here, γ represents the momentum term, typically set to 0.9.

IV. EXPERIMENTS

A. Experimental Environment & Dataset

TABLE I. EXPERIMENTAL ENVIRONMENT

Experimental Environment	**Version**
CPU	Intel Xeon E5-2698
GPU	NVIDIA Tesla V100 32G
Language	Python 3.8
Framework	Pytorch

The hardware configuration for this study consists of an Intel Xeon E5-2698 CPU, paired with four NVIDIA Tesla V100 GPUs. The system environment is Ubuntu 18.04, and the model is built using the PyTorch framework with Python version 3.8.

The dataset used in this study is based on DIOR and UCAS-AOD, which have been fused and processed together for joint training to enhance the model's robustness.

The performance of the model is quantified using accuracy and success rate. Accuracy is defined as the proportion of frames in which the target's center position error (Δ) falls below a specified threshold, compared to the total number of frames. In previous studies, this threshold is typically set at 20. The method for determining accuracy can be expressed as follows:

$$Precision = \frac{Count(\Delta < 20)}{Count(\Delta_{(all)})} \tag{6}$$

Where Δ is:

$$\Delta = \sqrt{\left(x_p - x_r\right)^2 + \left(x_p - y_r\right)^2} \tag{7}$$

The success rate refers to the ratio of the number of frames in which the overlap between the successfully tracked target detection box and the ground truth box exceeds a predefined threshold to the total number of frames. It measures the intersection-over-union (IoU) of the computed box with the ground truth box, and the formula is as follows:

$$IoU = \frac{Area\ of\ Interserction\ of\ two\ boxes}{Area\ of\ Union\ of\ two\ boxes} \tag{8}$$

$$Sucess = \frac{Count(IoU > 0.5)}{Count(IoU_{all})} \tag{9}$$

B. Experimental Results

In this study, we utilized a custom-built test set specifically designed for the evaluation of our proposed tracking method. The results of our experiments are illustrated in Figure 7, which showcases representative examples of the tracking performance achieved by our approach.

Figure 7. Experimental results. The red part is the model result and the green part is the real frame.

As depicted in the figure, the proposed method exhibits remarkable efficacy in tracking small targets within remote sensing images. Notably, it maintains a high level of accuracy in both tracking and recognition, even under challenging conditions such as occlusion. This level of robustness is essential in real-world scenarios where targets might be partially hidden by environmental elements or other objects.

To further validate the effectiveness of our approach, we conducted a comparative analysis against five state-of-the-art tracking models from recent years: SiamRPN [13], SiamRPN++ [1], SiamMask [15], SiamBAN [23], and SiamCar [24]. These models were selected based on their prominence in the field of object tracking, ensuring a comprehensive evaluation of our method's performance. The results are summarized in the table below:

TABLE II. THE SUCCESS RATE AND ACCURACY OF THIS METHOD ARE COMPARED WITH THE SOTA METHOD. THE RED VALUE IN THE TABLE IS HIGHEST, AND GREEN VALUE IS SECOND HIGHEST.

Models	Years	*Precision*	*Success*
SiamRPN	2018	0.753	0.342
SiamRPN++	2018	0.435	0.261
SiamMask	2019	0.569	0.278
SiamBAN	2020	0.784	0.497
SiamCar	2022	0.769	0.502
Ours	-	0.803	0.549

The experimental results clearly indicate that the enhancements introduced in this study have led to a significant improvement in the model's performance. Specifically, our proposed method achieves an impressive accuracy increase of 1.9%, reflecting a more precise capability in target tracking within remote sensing images. Additionally, we observe a notable improvement in the success rate, which has risen by 4.7%.

This increase in success rate signifies a substantial advancement in the model's ability to consistently and reliably track targets, even under challenging conditions such as occlusion and varying backgrounds. The enhancements not only demonstrate the effectiveness of the modifications made to the network architecture but also underscore the potential of our approach in real-world applications where high accuracy and robustness are paramount.

V. CONCLUSION

This study investigates target tracking methods for remote sensing imagery and presents several innovations by the authors. Based on the SiamRPN++ framework, a series of enhancements were introduced. Firstly, the C3Minus and CA modules were incorporated into the backbone network, significantly improving feature fusion and extraction capabilities. These additions allow the network to capture richer feature information when processing remote sensing images, resulting in enhanced tracking accuracy, especially in challenging scenarios with complex backgrounds and changing target appearances. Additionally, this study introduces the novel RestSwinT module, which combines the strengths of the Swin Transformer and ResNet to bolster the network’s spatiotemporal feature extraction capabilities. In target tracking tasks, the effective integration of spatiotemporal information enables the network to capture dynamic target changes more accurately. By incorporating the RestSwinT module, the network achieves more effective spatiotemporal feature fusion, further enhancing target identification and tracking performance.

With ongoing advancements in deep learning and computer vision, future research in this domain could explore methods for the effective integration of multimodal data (such as infrared and synthetic aperture radar imagery) to enhance tracking accuracy and reliability. Optimizing network architectures and algorithms to enable more efficient real-time tracking performance is also a promising direction.

REFERENCES

[1] Li, Bo, et al. "Siamrpn++: Evolution of siamese visual tracking with very deep networks." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.

[2] Hou, Qibin, Daquan Zhou, and Jiashi Feng. "Coordinate attention for efficient mobile network design." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

[3] Rakhlin, A. "Convolutional neural networks for sentence classification." GitHub 6 (2016): 25.

[4] Liu, Ze, et al. "Swin transformer: Hierarchical vision transformer using shifted windows." Proceedings of the IEEE/CVF international conference on computer vision. 2021.

[5] Kulikova, Maria V., and G. Yu Kulikov. "NIRK-based accurate continuous–discrete extended Kalman filters for estimating continuous-time stochastic target tracking models." Journal of Computational and Applied Mathematics 316 (2017): 260-270.

[6] Zhou, Huan, et al. "Adaptive unscented Kalman filter for target tracking in the presence of nonlinear systems involving model mismatches." Remote Sensing 9.7 (2017): 657.

[7] Habibi, Saeid. "The smooth variable structure filter." Proceedings of the IEEE 95.5 (2007): 1026-1059.

[8] Xuan, Shiyu, et al. "Object tracking in satellite videos by improved correlation filters with motion estimations." IEEE Transactions on Geoscience and Remote Sensing 58.2 (2019): 1074-1086.

[9] Liu, Yaosheng, et al. "Object tracking in satellite videos based on correlation filter with multi-feature fusion and motion trajectory compensation." Remote Sensing 14.3 (2022): 777.

[10] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[11] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[12] Bertinetto, Luca, et al. "Fully-convolutional siamese networks for object tracking." Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14. Springer International Publishing, 2016.

[13] Li, Bo, et al. "High performance visual tracking with siamese region proposal network." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[14] Zhu, Zheng, et al. "Distractor-aware siamese networks for visual object tracking." Proceedings of the European conference on computer vision (ECCV). 2018.

[15] Wang, Qiang, et al. "Fast online object tracking and segmentation: A unifying approach." Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. 2019.

[16] Xu, Yinda, et al. "SiamFC++: Towards robust and accurate visual tracking with target estimation

guidelines." Proceedings of the AAAI conference on artificial intelligence. Vol. 34. No. 07. 2020.

[17] Yan, Bin, et al. "Lighttrack: Finding lightweight neural networks for object tracking via one-shot architecture search." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[18] Cao Z, Huang Z, Pan L, et al. TCTrack: Temporal contexts for aerial tracking[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 14798-14808.

[19] Zhou, Li, et al. "Joint visual grounding and tracking with natural language specification." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023.

[20] Hong, Lingyi, et al. "Onetracker: Unifying visual object tracking with foundation models and efficient tuning." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.

[21] Mao, Anqi, Mehryar Mohri, and Yutao Zhong. "Cross-entropy loss functions: Theoretical analysis and applications." International conference on Machine learning. PMLR, 2023.

[22] Barron, Jonathan T. "A general and adaptive robust loss function." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.

[23] Chen, Zedu, et al. "Siamese box adaptive network for visual tracking." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[24] Guo, Dongyan, et al. "SiamCAR: Siamese fully convolutional classification and regression for visual tracking." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

Research and Design of an Intelligent IoT Monitoring System for Coal Mine Gas Based on the Fuzzy-PID Algorithm

Shengquan Yang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: xaitysq@163.com

Zhengxin Zhang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 478773436@qq.com

Jun Zhang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 2970784844@qq.com

Ruixin Ji
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 2119867770@qq.com

***Abstract*—To enhance the safety and efficiency of coal mine gas monitoring, this study develops an intelligent Internet of Things (IoT) monitoring system incorporating a Fuzzy-PID control algorithm. The system is structured into four layers—sensing, network transmission, control service, and mobile application—ensuring real-time data acquisition, stable transmission, intelligent processing, and remote monitoring. The Fuzzy-PID algorithm dynamically adjusts control parameters to improve response time and accuracy under nonlinear and uncertain conditions. Simulation experiments validate the system's performance, comparing traditional PID, Fuzzy, and Fuzzy-PID control strategies. Results indicate that the traditional PID algorithm achieves a response time of 2.0 s but exhibits oscillations of ±0.1 concentration units. The Fuzzy control algorithm stabilizes gas concentration within 4.0 s with deviations below ±0.05 units. The proposed Fuzzy-PID algorithm achieves an optimal balance, stabilizing gas concentration within 2.5 s with deviations reduced to less than ±0.03 units. These improvements enhance mine safety by reducing gas concentration fluctuations and providing real-time risk alerts. Practical deployment in a coal mining enterprise confirms the system's capability in reducing manual intervention by 30% and improving early warning accuracy by 25%, demonstrating its potential for intelligent mine development.**

Keywords-Fuzzy PID; Coal Mine Gas; Internet of Things; Monitoring System

I. INTRODUCTION

Abnormal fluctuations in coal mine gas concentration are a significant hidden danger leading to safety accidents, especially during deep mining. Gas concentration is often influenced by complex factors such as mining activities and ventilation conditions, exhibiting nonlinear, time-varying, and uncertain characteristics traditional gas monitoring systems primarily use fixed-parameter PID control algorithms to monitor gas concentration and control ventilation [1]. However, due to the complexity and dynamic nature of the coal mine environment, the performance of fixed-parameter PID algorithms is often limited, making it difficult to adapt quickly to complex conditions. This results in regulation delays and insufficient control precision [7].

In recent years, coal mine gas monitoring technology in China has made significant progress. Traditional single-point monitoring has gradually been replaced by multi-point, distributed monitoring systems, which rely on IoT technology to achieve real-time collection of parameters such as gas concentration, temperature, humidity, and wind speed [2]. However, most existing systems still use traditional PID control algorithms, which struggle to adapt to complex working conditions [8].

Developed countries have been conducting research on coal mine gas monitoring technology earlier, leveraging advanced industrial automation and IoT technology [3]. They have built real-time monitoring systems with multi-sensor integration. For example, countries such as the United States and Germany have achieved efficient monitoring and precise control of gas concentration in mines through a combination of distributed monitoring and intelligent control algorithms. While traditional PID algorithms have been widely applied in IoT systems abroad, there are still challenges related to high sensor energy consumption and inadequate real-time performance and stability of control algorithms under complex conditions [4]. Especially for dynamic multi-variable conditions, existing methods still fail to meet practical demands [5].

In recent years, fuzzy control (Fuzzy Control) technology has emerged as an effective solution for controlling complex dynamic systems, thanks to its adaptability and robustness for nonlinear systems [11]. By introducing fuzzy logic, fuzzy control dynamically adjusts the parameters of the PID controller based on real-time monitoring data and system status, ensuring excellent control performance under varying conditions. Applying the Fuzzy-PID algorithm to coal mine gas monitoring systems can effectively overcome the limitations of traditional control methods, enabling precise gas concentration control and efficient management [17].

This study proposes to research and develop the “Intelligent IoT Monitoring System for Coal Mine Gas Based on the Fuzzy-PID Algorithm,” with the following significant implications:

1) Improved Precision and Robustness in Gas Concentration Control: By introducing the Fuzzy-PID algorithm, the PID controller parameters can be dynamically adjusted, allowing the system to adapt to the nonlinear variations in gas concentration. This enables rapid response and stable control, significantly reducing the risk of abnormal gas concentration fluctuations and enhancing mine safety.

2) Deep Integration of IoT Technology: By combining IoT architecture, a multi-source sensor network will be constructed to achieve real-time collection and intelligent fusion of parameters such as gas concentration, temperature, and humidity. Through the integration of edge computing and cloud computing, the system’s data transmission efficiency and overall intelligence will be improved, creating a highly efficient coal mine gas monitoring platform.

3) Promoting Digital Transformation in the Coal Mining Industry: The research outcomes will not only provide new technical solutions for coal mine safety monitoring but also accelerate the industry’s shift toward digitalization and intelligence. The combination of intelligent monitoring and dynamic control technologies will contribute to achieving the dual goals of safe production and green development in coal mining.

4) Addressing Dynamic Control Challenges in Complex Conditions: This study will design optimized Fuzzy-PID control strategies tailored to complex dynamic conditions. By refining algorithms and conducting experimental validation, the system's stability and reliability under extreme conditions will be enhanced, providing technological support for addressing dynamic changes in the coal mining industry.

This research holds both academic and practical value, offering direct technical support for production while ensuring safe operations in coal mines. It will also enhance the level of intelligence in the industry, contributing significantly to the safe and sustainable development of coal mines.

II. OVERALL DESIGN

Based on the complexity of the mine environment and the high requirements for real-time performance and safety in gas monitoring, the overall architecture of the intelligent IoT monitoring system for coal mine gas, as shown in Figure 1, has been designed. The system is divided into four layers: the perception and execution layer, the network transmission layer, the control service layer, and the mobile application layer, ensuring modular functionality and efficient collaboration. The perception and execution layer is responsible for accurately collecting environmental data and executing control commands. The network transmission layer ensures stability and real-time performance in data transmission through multi-protocol communication. The control service layer, with the Fuzzy-PID algorithm as its core, achieves intelligent analysis and dynamic regulation. The mobile application layer provides convenient monitoring and remote control interfaces. The layered design clearly defines functionalities, reduces system complexity, and enhances scalability and maintainability. It also ensures real-time and secure data transmission and processing, thereby meeting the requirements for intelligence and reliability in coal mine gas monitoring systems.

A. Perception Layer

The Perception Layer serves as the foundation of the system, responsible for real-time collection of various physical parameters in the mine environment. These include gas concentrations (CH4, CO2, CO), oxygen concentration (O2), temperature and humidity, pressure, and wind speed. This data is critical for reflecting the real-time status of the mine environment and serves as the core basis for subsequent intelligent control and early warning [6]. The perception layer also performs preliminary processing of sensor data to eliminate interference, improve data quality, and standardize it into a unified data format.

The perception layer is composed of various high-precision sensors. For example, gas sensors utilize electrochemical or infrared detection technology to ensure stability under high-temperature and high-humidity conditions. Oxygen sensors use long-life sensing elements for continuous monitoring, while temperature, humidity, pressure, and wind speed sensors employ fast-response chips to adapt to dynamic changes in the mining environment. All sensors are centrally managed through Data Acquisition Terminals (DAT), which optimize collected data using filtering and calibration algorithms and transmit it via the Modbus protocol.

The perception layer interacts with the network layer through RS485 buses or wireless transmission modules (e.g., WiFi). Sensor data is uploaded to the network layer via the communication interface of the data acquisition terminal, while control commands from the control service layer are received simultaneously. For instance, real-time gas sensor data uploaded to the network transmission layer can trigger the activation or adjustment of ventilation equipment.

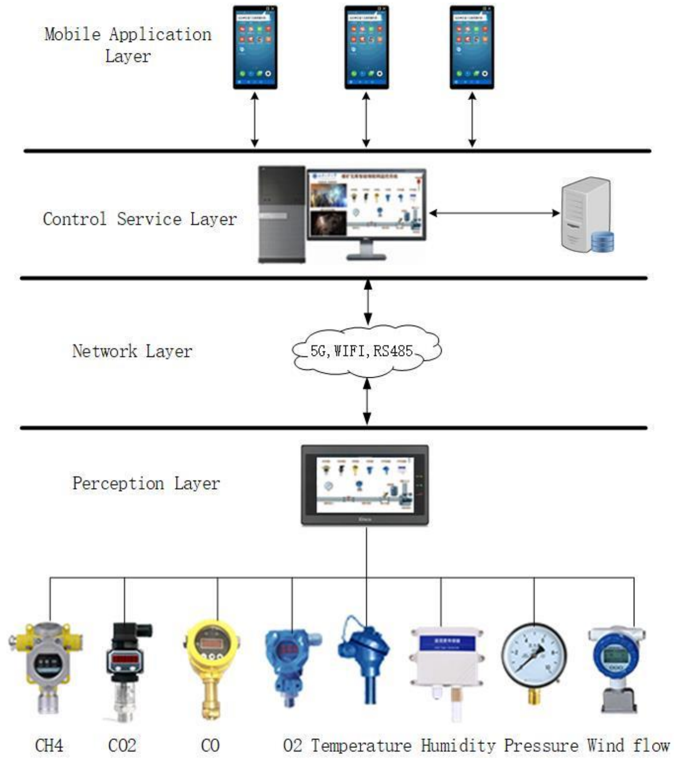


Figure 1. The Overall Architecture Diagram of the Intelligent IoT Monitoring System for Coal Mine Gas

B. Network Layer

The network layer serves as a data transmission bridge between the perception layer and the control service layer, responsible for the efficient transmission of sensor data and the reliable delivery of control commands. By utilizing multiple communication protocols such as 5G, WiFi, and RS485, the network layer ensures real-time uploading of environmental data from the mine and accurate transmission of control strategies generated by the control service layer to execution devices, maintaining the dynamic operation of the system [9].

The network layer supports a multi-protocol, multi-node communication structure. Wired communication (RS485) is used for short-distance transmission, suitable for reliable data delivery between mine sensor nodes. WiFi enables medium-distance data transmission, facilitating mobile device access, while 5G supports high-bandwidth, long-distance data transmission, accommodating the complex topography and large data volumes of the mine. Additionally, the network layer integrates data encryption technologies (e.g., TLS) and firewalls to ensure the security of data transmission.

The network layer interacts with the perception layer and control service layer through standardized protocols such as MQTT and HTTP. The perception layer uploads sensor data to network nodes via RS485 or WiFi, and the network layer transmits the data to the control service layer in JSON format. Simultaneously, the control service layer issues control commands through the network layer, leveraging the low-latency capabilities of the 5G network to meet the real-time control requirements of the mine.

C. Control service layer

The control service layer is the core of the system, responsible for data analysis, storage, and the generation of control strategies. Utilizing a combination of cloud platforms and local servers, the control service layer runs the Fuzzy-PID algorithm to intelligently regulate gas concentrations and environmental parameters in the mine. Additionally, it stores historical data and performs big data analysis to support forecasting of mine environment trends and the optimization of control strategies.

The control service layer consists of high-performance servers and database systems. Cloud servers execute the Fuzzy-PID algorithm, dynamically adjusting PID parameters through fuzzy logic to achieve precise control of ventilation equipment. The database system combines relational databases (e.g., MySQL) and NoSQL databases (e.g., MongoDB) to store both structured and unstructured data. Furthermore, the control service layer features intelligent early warning capabilities, allowing it to trigger alarms based on real-time data and notify mine management personnel to take action.

The control service layer interacts with the network layer via RESTful APIs, receiving standardized real-time data and returning optimized control instructions. Its interface with the mobile application layer supports real-time monitoring data queries and the pushing of early warning notifications [10]. Through these interfaces, users can access historical data and trend analyses. Additionally, the control service layer includes a command parsing module, which translates control strategies into executable commands for the devices in the perception layer.

D. Mobile application layer design

The mobile application layer provides users with real-time monitoring, remote control, and anomaly warning functions, serving as the system's front-end interaction interface. Through a user-friendly interface, the mobile application layer allows users to view mine environment data such as gas concentration and temperature/humidity anytime and remotely operate mine ventilation equipment via control interfaces.

The mobile application layer consists of both PC and mobile (Android/iOS) platforms. The PC platform offers advanced data visualization features, including multidimensional trend analysis and comparative charts, while the mobile platform focuses on lightweight design, providing real-time monitoring and quick operation capabilities. Additionally, the mobile application layer integrates push notification services, promptly notifying users of gas concentration anomalies via alerts or SMS and recommending appropriate measures.

The mobile application layer interacts with the control service layer through the HTTPS protocol, retrieving real-time environmental data and alarm information while uploading user-issued remote control commands. Utilizing WebSocket technology, the mobile application layer achieves bidirectional real-time communication with the control service layer, ensuring users receive instant updates on mine environment data and equipment status [12]. Furthermore, the mobile platform features access control through interfaces to restrict user permissions, ensuring the security of the system.

III. DESIGN OF NODES IN THE INTELLIGENT IOT MONITORING SYSTEM FOR COAL MINE GAS

A. Hardware design of sensor nodes

To achieve efficient, precise, and reliable data acquisition and processing in the complex mine environment, the sensor node hardware structure, as shown in Figure 2, has been designed. Through modular design, each functional module (such as sensors, filtering, A/D conversion, processors, communication, and power management) is clearly delineated, ensuring the system's adaptability and scalability. The filtering module eliminates signal noise, the A/D conversion module ensures high-precision signal digitization, the processor handles data integration and analysis, the communication module guarantees reliable long-distance data transmission, and the power management module provides stable power supply support. This architecture not only meets the multi-parameter monitoring requirements of mines but also optimizes system performance through inter-module collaboration, enhancing anti-interference capabilities, adapting to harsh mining environments, and ensuring stable operation and reliable communication of the system.

1) Sensor module

The sensor module is the core of the IoT sensor node for coal mine gas monitoring, responsible for collecting key parameters in the mine, such as gas concentrations (e.g., $CH4$, $CO2$, CO), oxygen concentration ($O2$), temperature and humidity, pressure, and wind speed. These sensors convert physical signals into electrical signals, providing reliable environmental data for subsequent modules. For example, the CH4 sensor can use the Figaro TGS2611, which employs non-dispersive infrared (NDIR) technology and can accurately measure methane concentrations in high-humidity and dusty mine environments, with features like low power consumption and long lifespan. For CO2 monitoring, the Senseair S8-0048 is recommended for its high accuracy and suitability for applications with a wide concentration range. For CO detection, the Alphasense CO-B4 sensor, based on electrochemical principles, offers high sensitivity and resistance to interference from other gases. The DHT22 is a suitable choice for temperature and humidity monitoring, featuring fast response and high accuracy, making it ideal for dynamic environments. These sensors output different electrical signals, such as voltage or current signals, based on the environmental parameters they collect, providing input for subsequent signal processing modules.

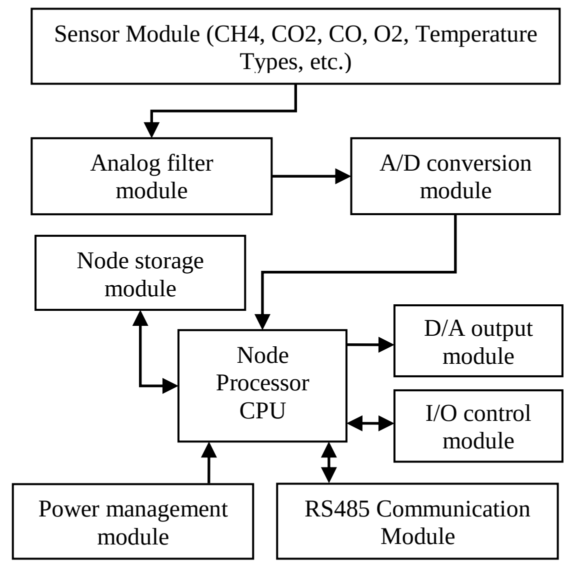


Figure 2. Functional Block Diagram of the Hardware Composition of IoT Sensor Nodes for Coal Mine Gas Monitoring

2) Analog Filtering Module

The analog filtering module is a critical component of signal processing, primarily responsible for removing noise and interference from sensor signals to ensure that the subsequent A/D conversion module receives more stable and accurate signals. Due to the high levels of electromagnetic interference and vibration noise in mining environments, the filtering module requires specialized design. Typically, the analog filtering module employs low-pass filters to suppress high-frequency interference signals. For example, an RC low-pass filter can be used to filter out high-frequency noise with a simple resistor-capacitor combination, or an active filter circuit built with an operational amplifier like the LM741 can enhance signal processing precision. For higher performance requirements, the OPA134 from TI, featuring low noise and a wide bandwidth, is suitable for high-precision sensor filtering needs. Filtered signals have a higher signal-to-noise ratio, providing a reliable data foundation for subsequent

A/D conversion. Additionally, the filtering module can customize filtering parameters based on the sensor type, such as adjusting the cutoff frequency to suit the response characteristics of gas or temperature and humidity sensors, further optimizing system performance.

3) A/D Conversion Module

The A/D conversion module converts analog signals from sensors into digital signals for data analysis and processing by the node processor. This module must ensure sufficient resolution and sampling rate to meet the multi-parameter monitoring needs in the complex mine environment. For instance, TI's ADS1115 is recommended—a 16-bit high-precision ADC chip with four-channel input and low power consumption, suitable for the energy-constrained mining environment. The ADS1115 achieves a sampling rate of up to 860 SPS (samples per second), which meets the monitoring requirements for rapidly changing parameters such as gas concentration and temperature/humidity. Additionally, the ADS1115 supports the I2C communication protocol, offering strong compatibility with node processors and ease of integration. For scenarios requiring higher precision, a 24-bit ADC chip such as the AD7190 from Analog Devices can be used, ideal for converting sensor signals that demand higher resolution. The selection of the A/D conversion module should consider factors like the dynamic range of sensor output signals, system real-time requirements, and energy constraints to achieve optimal performance.

4) Node Processor Module

The node processor module is the "brain" of the entire sensor node, responsible for integrating, processing, and analyzing sensor data, as well as interacting with upper-layer systems via the communication module. ARM Cortex-M series processors, such as the STM32F103, are recommended for their low power consumption, high performance, and rich peripheral interfaces, making them well-suited for data processing tasks in complex mining environments. The STM32F103 supports various communication protocols (e.g., I2C, SPI, and UART), enabling seamless integration with A/D conversion and communication modules. Additionally, this processor can run simple data correction algorithms, such as linearization or temperature and humidity compensation, to enhance sensor data accuracy. For scenarios requiring higher processing capabilities, the ESP32 processor is a suitable option, as it integrates WiFi and Bluetooth modules, supporting complex network communication and real-time data processing. The node processor also requires a low-power mode design to extend battery life in unmanned mining scenarios.

5) RS485 Communication Module

The RS485 communication module serves as the data transmission channel between sensor nodes and upper-layer systems. It is characterized by strong anti-interference capability and long transmission distances, making it highly suitable for the complex communication environments of mines. For instance, the MAX485 chip is recommended, a low-power, high-reliability RS485 transceiver capable of stable communication over distances up to 1200 meters. RS485 communication modules transmit data using differential signals, effectively resisting electromagnetic interference in mines. Additionally, RS485 modules support multi-node communication, allowing up to 32 devices to connect to a single bus, facilitating large-scale sensor node network deployment. This module is typically connected to the processor via a UART interface, responsible for uploading sensor data to the control service layer and receiving control commands from the upper layer. For example, the MAX485 can efficiently transmit gas concentration data, providing critical support for controlling ventilation fan operations.

6) Power Management Module

The power management module provides stable power supply to all parts of the sensor node, ensuring reliable operation in harsh environments. The AMS1117 voltage regulator chip is recommended to step down the commonly used 12V or 24V DC voltage in mines to 3.3V or 5V to meet the power supply needs of sensors and processors. To handle power outages or interruptions, backup batteries such as lithium battery packs can be included, with a charging management module (e.g., TP4056) for battery charging and management. The power management module should also include a voltage monitoring circuit to warn the processor when the power supply voltage is insufficient, ensuring that critical data is saved, and the system shuts down safely. This design extends the node's battery life, making it particularly valuable in unmanned mining environments.

B. Software design of sensor nodes

The microcontroller program design of the sensor node is shown in Figure 3. It adopts a modular design, where multiple independent subprograms collaborate to perform functions such as the collection, processing, storage, and communication of coal mine gas data. After powering on, the program first initializes parameters and then controls the sensors to collect real-time gas concentration data from the environment. The collected data undergoes filtering and noise reduction before being stored to ensure its accuracy and continuity. The entire process is efficient and stable, meeting the stringent real-time requirements of coal mine gas monitoring.

During the data processing phase, the program uses built-in algorithms to generate control instructions, which are executed via I/O or D/A interfaces to precisely control external devices (e.g., alarms). Additionally, the node provides LCD display and keypad functions, enabling on-site operators to view data or adjust operating parameters in real-time. The RS485 communication module ensures reliable data upload and inter-node communication, significantly enhancing the system's overall connectivity and resistance to interference.

The pseudocode for the algorithm implementation of this program is as follows:

```
int main() {
  // Step 1: Power on and initialization
  Power_On();
  Initialize_Parameters();
  while (1) {
    // Step 2: Data acquisition
    float gas_data = Acquire_Data();
    if (gas_data == -1) {  // Error handling for failed acquisition
      Log_Abnormal_Condition(gas_data);
      Trigger_Alarm();
      Communicate_Emergency();
      Recover_System();
      continue;  // Restart loop after error handling
    }
    // Step 3: Data filtering and storage
    float filtered_data = Filter_Data(gas_data);
    Store_Data(filtered_data);
    // Step 4: Algorithm control
    int control_signal = Algorithm_Control(filtered_data);
    Output_Control(control_signal);
    // Step 5: Key input handling
    if (Key_Pressed()) {
      Process_Key_Input();
    }
    // Step 6: RS485 communication
    if (RS485_Available()) {
      Send_Data(filtered_data);
      Receive_Commands();
    }
    // Step 7: LCD display update
    Update_LCD_Display(filtered_data);
    // Step 8: Exception detection
    if (Check_Abnormal(filtered_data)) {
      Log_Abnormal_Condition(filtered_data);
      Trigger_Alarm();
      Communicate_Emergency();
      Recover_System();
      continue;  // Restart loop after handling abnormal conditions
    }
  }
  return 0;
}
```

The greatest advantage of the program lies in its robustness and safety. With built-in anomaly detection and handling modules, it can promptly identify and respond to abnormal gas concentration situations, triggering alarms or recording fault data. This design not only ensures the reliability and scalability of the coal mine monitoring system but also provides strong technical support for safe coal mine production.

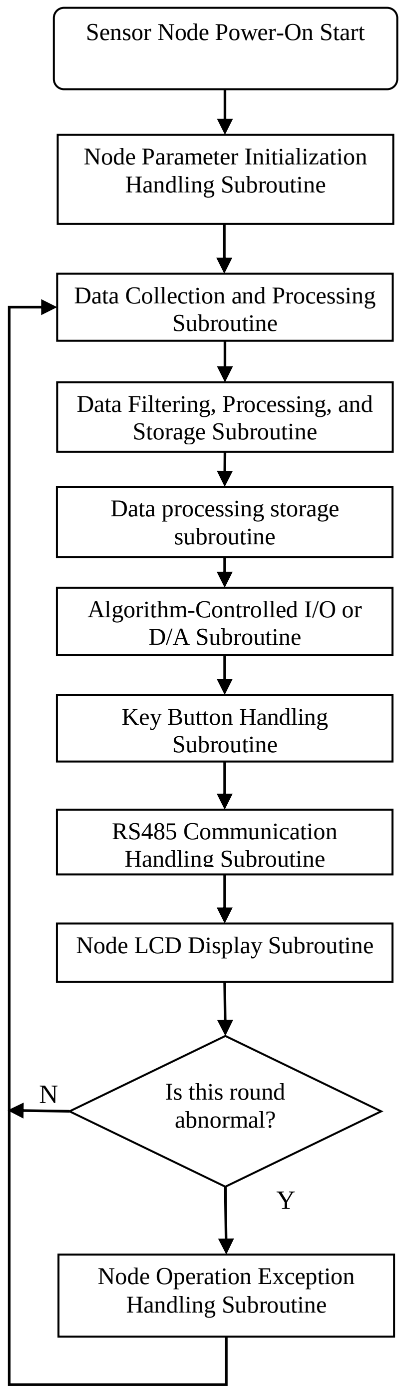


Figure 3. Flowchart of the Microcontroller Software Design for Sensor Nodes

IV. DESIGN OF CONTROL SERVICE LAYER SOFTWARE IN THE COAL MINE GAS INTELLIGENT IOT MONITORING SYSTEM

Based on the principles of modularity and layered design, the software functions of the control service layer are divided into the following modules: system user management module, RS485 communication interface module, data recording and storage module, process screen display module, dynamic curve display module, system logic control module, and Fuzzy-PID algorithm control module, as shown in Figure 4. These modules are responsible for core functionalities such as user access management, data transmission, historical record storage, monitoring interface display, data trend analysis, system operation logic processing, and dynamic parameter regulation.

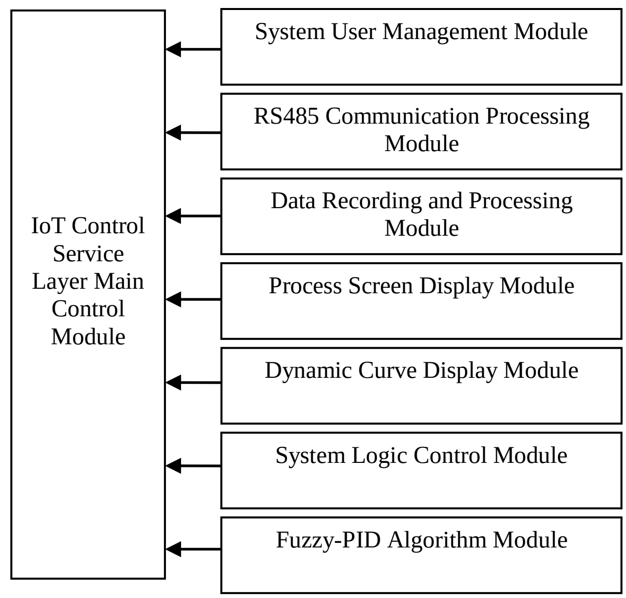


Figure 4. Function Module Diagram of the Control Service Layer in the IoT Monitoring System

Through the collaborative division of labor among these modules, the system can efficiently monitor coal mine gas concentration, perform real-time regulation, and respond to anomalies. The modular design ensures low coupling and high reusability between functional modules, enhancing system flexibility while facilitating future feature expansion and system maintenance.

A. System User Management Module

The system user management module is responsible for creating, managing, and assigning permissions to user accounts. Its main functions include user authentication, account creation and deletion, password management, and permission level control (e.g., administrators, regular users, maintenance personnel). The module uses encryption algorithms such as SHA-256 to securely store user passwords, ensuring data security. By defining user roles, it restricts access to sensitive data and high-privilege operations, enhancing overall system security. Additionally, the module supports logging features to track user operation history, providing data support for audits and troubleshooting. Technical specifications include the maximum number of supported users (e.g., 1,000), the number of permission levels (e.g., 5 levels), and the storage capacity for operation logs (e.g., 10GB).

B. RS485 Communication Interface Module

The RS485 communication interface module facilitates communication between the control service layer and sensor nodes. Its core functions include real-time data transmission, communication status detection, data verification (e.g., CRC checks), and fault recovery mechanisms. The module supports a maximum communication distance of 1,200 meters and a maximum data transfer rate of 10 Mbps, meeting the data transmission requirements of complex coal mine environments. Utilizing a master-slave communication model ensures reliability and stability in data transfer. Additionally, the module supports concurrent connections with up to 32 slave nodes, making it suitable for large-scale sensor deployment scenarios. The module also integrates a communication interruption detection mechanism, enabling quick retries or alerts in case of communication failure.

C. Data Recording and Storage Module

The data recording and storage module is responsible for real-time data logging and storage. It supports data types such as gas concentration, equipment operating status, and alarm information, using relational databases like MySQL for storage. Its functions include real-time data writing, periodic backups, historical data retrieval, and report generation. The module supports a data recording frequency of 10 times per second, with scalable storage capacity up to terabyte levels. Data compression technology optimizes storage space utilization. Additionally, the module supports data backup and recovery, enabling rapid restoration in case of data loss.

D. Process Screen Display Module

The process screen display module provides a visual interface for dynamically presenting system operation status and monitoring data. Interface elements include gas concentration trend charts, equipment status icons, and alarm notifications. The module supports a resolution of 1920×1080 and a refresh rate of 60 Hz, ensuring clear and smooth visuals. Users can adjust display content through touch operations, with features such as multi-language support and customizable interface layouts. Developed using HTML5 and JavaScript, the module supports remote access and cross-platform display, providing operators with an intuitive monitoring tool.

E. Dynamic Curve Display Module

The dynamic curve display module is primarily used for trend analysis and historical playback of monitoring data. Its functions include real-time curve plotting, historical data playback, curve zooming and panning, and comparative analysis. The module supports up to 10 curves displayed simultaneously with an update frequency of 1 Hz. By employing efficient data caching algorithms, it ensures responsive performance even when processing large datasets. Users can customize display parameters (e.g., time range and data source) through the interactive interface, facilitating personalized analysis. This module is particularly useful for evaluating equipment performance and tracing anomalous data.

F. System Logic Control Module

The system logic control module handles the core logic processing of system operations, including comprehensive analysis of sensor node data, status monitoring, and the generation of control commands. Its functions include receiving and parsing sensor data, generating action instructions based on control strategies (e.g., starting fans or closing valves), distributing control signals, and coordinating the overall system operating status. The module supports flexible logic configuration, allowing control strategies to be adjusted according to specific needs (e.g., switching between automatic and manual modes). Technical parameters include a response time of less than 10 ms and support for up to 100 logic control rules, ensuring stable and efficient operation of complex systems.

G. Fuzzy-PID Algorithm Control Module

The Fuzzy-PID algorithm control module dynamically regulates critical system parameters. By incorporating fuzzy control algorithms, it achieves precise control in nonlinear and complex environments where traditional PID control falls short [20]. Its functions include fan speed control during gas concentration exceedance and system stability optimization. The module supports online adaptive parameter adjustment, with a sampling frequency of 50 times per second and control precision up to 0.1%. By optimizing the fuzzy rule base, the module can automatically adjust PID parameters based on real-time data, enhancing control effectiveness, especially in dynamic coal mine environments.

Together, these modules form the core framework of the control service layer, each performing its designated role while collaborating to deliver efficient, secure, and precise solutions for coal mine gas monitoring. The system logic control module, in particular, serves as the foundation of efficient system operation by handling data analysis, command generation, and status coordination.

The IoT system's mobile application layer software design emphasizes real-time performance, convenience, interactivity, and security. Seamlessly integrated with the control service layer, it provides users with functionalities such as gas concentration monitoring, alarm notifications, historical data queries, and remote operations. The mobile application supports multi-platform compatibility, featuring an intuitive and user-friendly interface, and ensures system security through identity authentication, data encryption, and access control. Real-time alarm notifications and intelligent notification mechanisms keep users informed of system status anytime and anywhere, making it an efficient mobile solution for coal mine gas monitoring. The mobile application layer software design shares similarities with the control service layer's functional modules, which are not reiterated here for brevity.

V. DESIGN OF CORE CONTROL ALGORITHMS FOR THE SYSTEM

The design philosophy of this system centers on intelligence, real-time performance, and adaptability. By integrating fuzzy control with traditional PID control, it overcomes the nonlinear and dynamic characteristics of complex coal mine environments, achieving precise gas concentration regulation. The core controller collects sensor data in real-time, dynamically adjusts PID parameters through fuzzy reasoning, optimizes control accuracy and response speed, and seamlessly integrates with the IoT system. It supports data analysis, alarm linkage, and remote monitoring, ensuring the safety and efficiency of coal mine operations.

A. Traditional PID Algorithm

Traditional PID control (Proportional-Integral-Derivative Control) is a classical automatic control algorithm widely used in industrial control systems. Its fundamental principle involves the coordinated action of three components: proportional (P), integral (I), and derivative (D) [17]. By analyzing the error between the control target and the actual output, it dynamically adjusts the control variable to gradually bring the system output closer to the desired value, achieving precise automatic control.

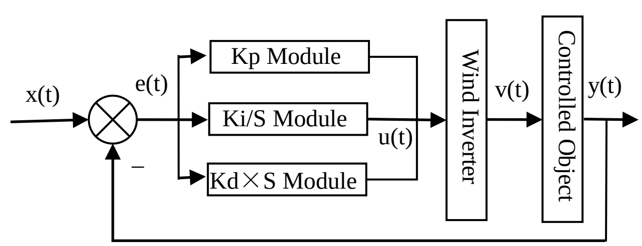


Figure 5. Traditional PID Control Process for Ventilation Variable Frequency Drives in Coal Mine Gas IoT

In coal mine gas IoT systems, variable frequency drives (VFDs) controlling ventilation fans often employ continuous control methods. The output current signal typically ranges from 4 to 20mA, and the general process of traditional PID control is illustrated in Figure 5.

The control principle is shown in equation (1).

$$u(t) = K_p[e(t) + \frac{1}{T_I}\int_0^t e(t)dt + T_D\frac{de(t)}{dt}] \quad (1)$$

Where Kp is the proportional coefficient; TI is the integral time constant; TD is the derivative time constant; e(t) is the error; and u(t) is the control variable.

After discretization, the discrete algorithm for positional PID is obtained, as shown in equation (2).

$$u(k) = K_p e(k) + K_I\sum_{i=0}^{k} e(i) + K_D[e(k) - e(k-1)] \quad (2)$$

The output u(k)u(k) of the PID controller is related to all past error signals, requiring the system to accumulate e(i), which leads to significant computational workload. Additionally, delays in external signal acquisition or interference faults may cause u(k) to oscillate significantly. Such scenarios often result in unsatisfactory control performance and, in some cases, can lead to serious accidents. Furthermore, the output u(k) of the controller corresponds to the actual position of the actuator; when the system experiences significant delays in signal acquisition, large changes in u(k) can cause drastic position changes in the actuator.

In practical control systems, an incremental PID algorithm can be used to mitigate these issues. Based on the previous formulas, it calculates the difference between two consecutive time points. The formula is shown as equation (3):

$$\begin{aligned}\Delta u(k) &= u(k) - u(k-1) \\ &= K_p[e(k) - e(k-1)] + K_I e(k) + K_D[e(k) - 2e(k-1) + e(k-2)]\end{aligned} \quad (3)$$

Where e(k) is the error value at the k-th sampling; e(k−1)e(k-1) is the error value at the (k−1)-th sampling; u(k) is the output of the regulator at the k-th sampling; Kp is the proportional coefficient; KI is the integral coefficient; and KD is the derivative coefficient. These coefficients can be expressed using the following formulas, as shown in equation (4):

$$K_I = K_P\frac{T}{T_I}, \quad K_D = K_P\frac{T_D}{T} \quad (4)$$

Based on the algorithm's structure, it is evident that the incremental PID algorithm has the following advantages over the positional PID algorithm:

1) Reduced Complexity

The incremental algorithm only depends on e(k)e(k), e(k−1)e(k-1), and e(k−2)e(k-2), eliminating the need for accumulation. This characteristic prevents integral saturation, enabling better control performance.

2) Seamless Manual-to-Automatic Transition

In positional control algorithms, transitioning from manual to automatic mode requires ensuring the computer's output matches the valve's original opening position to guarantee a disturbance-free switch, which complicates program design. In contrast, the incremental design only depends on the current error value and is independent of the valve's previous position, making it easier to achieve seamless manual-to-automatic transitions.

3) Reduced Impact of Faulty Actions

In the incremental algorithm, the computer only outputs the incremental value, minimizing the impact of faulty operations. Additionally, logical protection mechanisms can be implemented as needed to limit or block outputs during faults.

This flowchart illustrates the working principle of traditional PID control in coal mine gas IoT ventilation systems. The system calculates the error e(t)=x(t)−y(t) between the target value and the actual output. It then processes this error through proportional, integral, and derivative operations to generate a control signal u(t).

The control signal is input into the ventilation variable frequency drive (VFD), which adjusts the operating frequency of the fan, thereby changing the ventilation volume v(t) to regulate the state of the controlled object (e.g., gas concentration). Through closed-loop feedback control, the system continuously adjusts fan operations, ultimately making the actual output y(t) approach the target value x(t). This ensures precise and efficient gas concentration control while optimizing energy usage.

In practical applications of coal mine gas intelligent IoT monitoring systems, traditional PID algorithms face several challenges, including poor adaptability to nonlinear and dynamic environments, fixed parameters that struggle to accommodate environmental changes, sensitivity to noise and interference, and insufficient capability to handle multivariable coupling.

Moreover, in the complex coal mine environment, gas concentration often fluctuates drastically and is influenced by multiple factors. Traditional PID control struggles to achieve precise regulation and may experience response delays on devices with limited computational power. These shortcomings limit its control effectiveness in complex coal mine scenarios, necessitating optimization and improvement through the integration of fuzzy control or intelligent algorithms.

B. Fuzzy Algorithm

The Fuzzy Algorithm (Fuzzy Control Algorithm) is an intelligent control method based on fuzzy logic, widely applied in control scenarios involving nonlinear, uncertain, and complex systems. Its core concept is to use "fuzzy sets" and "fuzzy rules" to address problems that traditional control methods struggle to solve. By fuzzifying the system's input variables, applying a set of fuzzy rules for reasoning, and then defuzzifying the reasoning results into outputs, the algorithm achieves system control.

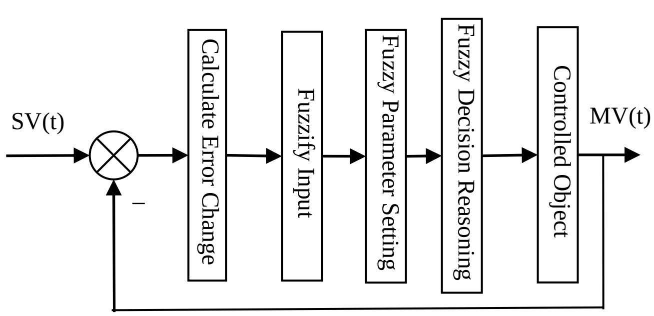


Figure 6. Fuzzy Control System Block Diagram

In coal mine gas intelligent IoT monitoring systems, the fuzzy algorithm can be utilized. Its core component is the Fuzzy Controller, and the basic control principle diagram is shown in Figure 6.

Based on the fuzzy control system framework shown in Figure 7, the working principle of fuzzy control can be described as follows:

The core of fuzzy control lies in error feedback. It begins by calculating the error e(t) and its rate of change Δe(t) between the set value SV(t) and the actual output PV(t), which reflects the system's deviation. These input parameters undergo fuzzification, converting them into linguistic variables (e.g., "positive large," "negative small") and describing their degree of fuzziness using membership functions. This step effectively handles nonlinear systems and uncertain environments.

During the fuzzy reasoning phase, the fuzzy controller applies a predefined fuzzy rule base (e.g., "If the error is large and the rate of change is small, then the control output should be small adjustments"). Combining the fuzzified inputs, it uses logical reasoning to generate fuzzy outputs. These rules, often derived from expert experience or system analysis, adapt flexibly to the dynamic changes of complex systems. The fuzzy reasoning output is a set of fuzzy values representing the membership degree of the control variable under different linguistic variables.

Finally, the fuzzy controller employs a defuzzification method to convert the fuzzy outputs into specific control signals MV(t), which are used to adjust the controlled object. After applying the control signal to the system, the closed-loop feedback continuously optimizes the control effect, gradually aligning the system output PV(t) with the desired value SV(t).

Fuzzy control does not require precise mathematical models and demonstrates excellent adaptability and robustness in environments with nonlinearity, multivariable coupling, and significant uncertainties, making it an effective complement to traditional control methods.

The Fuzzy Controller adopts a dual-input single-output control method, using temperature error e and error rate of change ec as input variables and UF as the output variable.

The fuzzy subsets are E=EC=UF={NB,NM,NS,ZE,PS,PM,PB}={NegativeBig,NegativeMedium,NegativeSmall,Zero,PositiveSmall,PositiveMedium,PositiveBig}. The domains are defined as e=ec=UF={−3,−2,−1,0,1,2,3}, or written as e:[−Xe,Xe], ec:[−Xec,Xec], uF:[−UF,UF].

The membership functions use triangular distribution functions. Based on practical experience, a set of reasoning rules is summarized and expressed in the form of if—then statements, resulting in a Fuzzy control rule table for the control variable UF, as shown in Table 1 [13].

(1) if E is NB and EC is NB then UF is PB

(2) if E is NB and EC is NM then UF is PB

……

(49) if E is PB and EC is PB then UF is NB

TABLE I. FUZZY CONTROL RULE TABLE

UF		E						
		NB	NM	NS	ZE	PS	PM	PB
EC	NB	PB	PB	PM	PM	PS	ZE	ZE
	NM	PB	PB	PM	PS	PS	ZE	NS
	NS	PM	PM	PM	PS	ZE	NS	NS
	ZE	PM	PM	PS	ZE	NS	NM	NM
	PS	PS	PS	ZE	NS	NS	NM	NM
	PM	PS	ZE	NS	NM	NM	NM	NB
	PB	ZE	ZE	NM	NM	NM	NB	NB

Based on the fuzzy rules, the fuzzy relationship is summarized. By applying Mamdani's fuzzy reasoning and composition operations, the membership degree μUF (E,EC) for the elements in the UF domain is obtained.

The defuzzification process uses the weighted average method to calculate the precise control value UF from the fuzzy controller. This control value adjusts the output voltage of the power regulator and regulates the temperature, effectively suppressing disturbances and enhancing the system's response speed and stability [14].

The design of a fuzzy controller can be carried out in the following steps [15]:

Step 1: Clearly define the input variables (Input Variable) and output variables (Output Variable) of the fuzzy controller (Fuzzy Controller).

Step 2: Design the control rule table (Control Rule Table) of the fuzzy controller based on the characteristics of the system.

Step 3: Determine the method for fuzzification and defuzzification (also called clarification) according to the fuzzy rule table (Rule Table).

Step 4: Select the domains of the input variables (Input Variable) and output variables (Output Variable) for the fuzzy controller, and determine parameters such as quantization factors and scaling factors for the Fuzzy Controller.

Step 5: Choose an appropriate sampling time (Sample Time) for the fuzzy control algorithm based on the system's application requirements, and obtain the discretized input data.

Step 6: Use the input data to run the fuzzy control algorithm module and obtain the output control data.

Theoretically, the more variables selected for the fuzzy controller, the higher the control precision of the system. However, when the number of variables is too large or the trends in input-output errors are unclear, the fuzzy control rules become overly complex, making control algorithms based on fuzzy reasoning and composition more challenging [16].

Furthermore, fuzzy control is a multivariable nonlinear disturbance control method. It inherently has uncertain static errors and may exhibit poor convergence, making stable control difficult to achieve.

C. Fuzzy-PID Algorithm

In coal mine gas intelligent IoT monitoring systems, traditional PID algorithms are widely used due to their simplicity and ease of implementation. However, they show insufficient adaptability when dealing with nonlinear, multivariable coupling, and dynamic changes in complex coal mine environments. Fuzzy algorithms excel at handling uncertainty and nonlinearity but require high system response speeds.

Combining the two into a Fuzzy-PID algorithm leverages their respective strengths: The Fuzzy algorithm dynamically adjusts PID parameters, enabling the system to optimize control performance under different operating conditions, while the PID algorithm provides high-precision execution control, ensuring rapid and stable system responses [17].

The Fuzzy-PID algorithm not only enhances the intelligence level of control but also improves the system's adaptability and robustness. Especially in complex gas environments, it effectively reduces overshoot and oscillation issues, delivering a more efficient and precise control solution for coal mine safety.

This system adopts a combination of Fuzzy control and PID control, retaining the advantages of PID control while incorporating the features of Fuzzy control [18]. The structural block diagram is shown in Figure 7.

When the system error e(t,k) is large, the focus is on Fuzzy control to improve system responsiveness. Conversely, when the system error e(t,k)is small, the emphasis shifts to PID regulation.

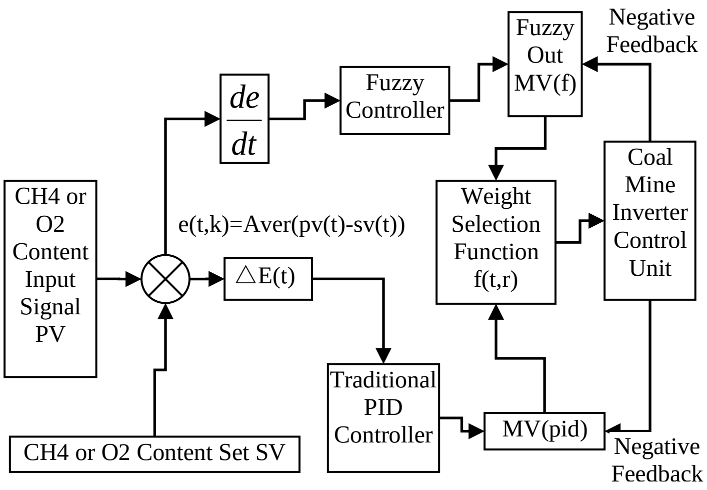


Figure 7. Coal Mine Gas Intelligent IoT Fuzzy-PID Control Schematic

The working process can be described as follows:

1) Error Calculation

The system receives the input signal PV(t) (CH4 or O2 content) and the setpoint SV(t) It calculates the error e(t)=AVER(PV(t)−SV(t)) and the rate of change of the error de/dtto obtain the core control parameters that reflect the system's state. These parameters serve as the input for the subsequent fuzzy control and traditional PID control.

2) Parallel Operation of Fuzzy Control and Traditional PID Control

Fuzzy Control: The fuzzy controller receives the error e(t) and the rate of change of the error de/dtas inputs. Based on the fuzzy rule base, it performs reasoning and generates the fuzzy control output MV(f). Fuzzy control is primarily responsible for dynamic adjustment and precise control under nonlinear operating conditions [19].

Traditional PID Control: The PID controller receives the error ΔE(t) as input and generates the traditional PID control signal MV(pid) based on proportional, integral, and derivative algorithms. PID control focuses on stable control and rapid response under linear operating conditions.

The fuzzy control output MV(f) and the traditional PID output MV(pid) are sent to the weight control selection function f(t,r), which is calculated at time t as shown in Equation (5). This function dynamically adjusts the weight ratio between fuzzy control and PID control based on the system's real-time operating conditions (e.g., degree of environmental change, error magnitude, etc.) [20]. By merging the control signals from both methods, the final integrated control output MV(Out) is generated.

$$M\varsigma(O\upsilon\tau)= M\varsigma(\pi\iota\delta)\times\rho+ M\varsigma(\phi)\times(1-\rho) \quad (5)$$

Where r is the weight coefficient. Based on this formula, it can be seen that r is actually a coordination factor that varies over time. The system can dynamically adjust the value of the coordination factor rr according to the real-time control deviation. This allows for the weighting of PID control and fuzzy control to be modified, fully leveraging the advantages of both fuzzy control and PID control while avoiding their respective shortcomings.

3) Control Execution and Feedback

The integrated control signal MVis sent to the coal mine ventilation variable frequency drive (VFD) control unit, which adjusts the fan speed and ventilation volume, thereby controlling the

CH4 or O2 concentration. Meanwhile, the system continuously monitors the actual output PV(t) via closed-loop feedback, compares it with the setpoint SV(t), and enters the next control cycle.

The core algorithm pseudocode is as follows:

```
int main() {
// Parameters
double PV, SV, error, prev_error = 0, error_rate;
double MV_fuzzy, MV_pid, MV;
double kp = 1.0, ki = 0.5, kd = 0.1;  // PID parameters
double fuzzy_weight = 0.7, pid_weight = 0.3;
while (1) {
    // Step 1: Read inputs
    PV = get_sensor_input();
    SV = get_setpoint();
    // Step 2: Calculate error and error rate
    error = SV - PV;
    error_rate = error - prev_error;
    // Step 3: Compute outputs
    MV_fuzzy = fuzzy_control(error, error_rate);
    MV_pid = pid_control(error, kp, ki, kd);
    // Step 4: Combine outputs with dynamic weights
    double state = fabs(error);  // Example state
calculation
    double weight = calculate_weight(state,
fuzzy_weight, pid_weight);
    MV = weight * MV_fuzzy + (1 - weight) * MV_pid;
    // Step 5: Output control signal
    set_actuator_output(MV);
    // Step 6: Update error
    prev_error = error;
    // Delay for the next control cycle
    delay(100);
  }
  return 0;
}
```

VI. ALGORITHM COMPARISON EXPERIMENT

To evaluate the performance of the PID, Fuzzy, and Fuzzy-PID algorithms for oxygen (O_2) monitoring in coal mine environments, a comprehensive two-stage experimental setup was designed to simulate realistic working conditions. The first stage involved a controlled laboratory simulation, where a sealed test chamber was constructed to replicate underground coal mine conditions. The chamber was equipped with industrial-grade O_2 sensors, a ventilation system, and a microcontroller-based real-time control platform to implement the three algorithms. A variable frequency drive (VFD) was integrated to adjust airflow dynamically, simulating oxygen concentration fluctuations caused by operational variations. The system was programmed to generate multiple test scenarios, including sudden O_2 drops (e.g., 15%) due to ventilation system failures and gradual concentration changes due to environmental instability. Performance metrics such as response time, steady-state error, and control stability were recorded to assess the adaptability of each algorithm under dynamic conditions.

In the second stage, the system was deployed in an active coal mining site to evaluate its effectiveness in real-world applications. Ten O_2 sensors were strategically installed in key ventilation zones to collect real-time data, which was transmitted via the IoT-based monitoring system to the control center. The industrial validation test was conducted over a 30-day period, during which the system automatically adjusted ventilation in response to varying O_2 levels. The experimental data originated from two main sources: first, calibrated industrial sensors continuously recorded O_2 concentrations to ensure high-resolution performance evaluation; second, historical datasets from coal mine safety organizations were integrated to create real-world disturbance scenarios, including rapid O_2 depletion events, long-term atmospheric trends, and sensor noise interference. The collected data was analyzed to compare the efficiency, reliability, and adaptability of the three algorithms, confirming the superiority of the Fuzzy-PID approach in ensuring stable and precise O_2 regulation within coal mine environments.

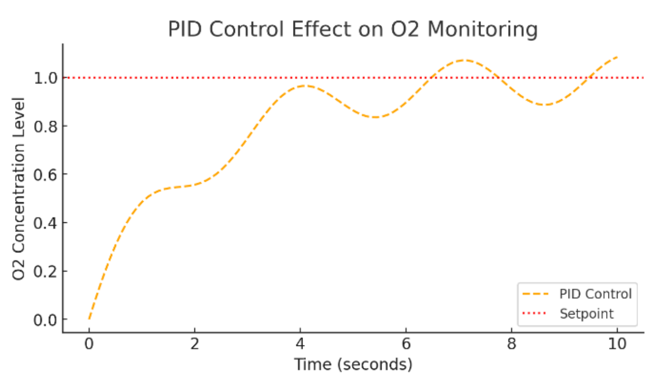


Figure 8. Experimental Control Effect Curve of the Traditional PID

The experiment aims to compare the three algorithms in terms of response time, stability, and ability to handle disturbances. PID is expected to show fast initial response but with oscillations, Fuzzy should perform better in non-linear scenarios with smoother results, and Fuzzy-PID is anticipated to combine the advantages of both, delivering optimal performance. By systematically analyzing their effectiveness, the experiment will validate which algorithm is best suited for intelligent O2 monitoring in coal mine safety systems. The experimental control effect curve of the traditional PID algorithm is shown in Figure 8.

The PID control algorithm demonstrates a fast initial response, reaching approximately 80% of the setpoint within 2 seconds, as seen from its curve. However, its oscillatory behavior around the setpoint, with variations of about ±0.1 concentration units, indicates a lack of stability in dynamic environments. This sensitivity to system disturbances or noise highlights its limitation in handling the complex, nonlinear nature of coal mine environments, where robust and adaptive control is required. The experimental control effect curve of the Fuzzy algorithm is shown in Figure 9.

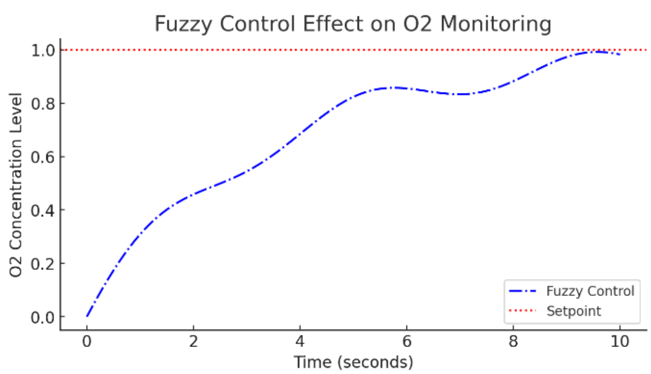


Figure 9. Experimental Control Effect Curve of the Fuzzy Algorithm

The Fuzzy control algorithm provides a smooth and stable response, with no significant oscillations after reaching the setpoint. It stabilizes within 4 seconds, slower than PID, but exhibits less than ±0.05 concentration unit deviation, demonstrating its robustness. However, the slower convergence rate suggests that Fuzzy control alone may not be suitable for scenarios requiring rapid corrective actions, such as sudden O2 level drops in coal mines. The experimental control effect curve of the Fuzzy-PID algorithm is shown in Figure 10.

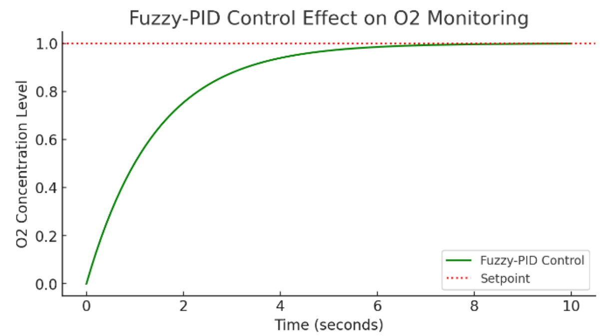


Figure 10. Experimental Control Effect Curve of the Fuzzy-PID Algorithm

The Fuzzy-PID algorithm exhibits a fast response, reaching 90% of the setpoint within approximately 2.5 seconds, demonstrating its efficiency in quickly adjusting to the desired oxygen level. Additionally, it stabilizes near the setpoint with minimal deviation of less than ±0.03 concentration units, ensuring precise and steady control. This combination of rapid response and stability makes the Fuzzy-PID algorithm particularly suitable for dynamic and complex environments like coal mine O2 monitoring, where both timely adjustments and robust performance are critical for maintaining safety and operational efficiency. The comparative experimental control effect curve of the three algorithms is shown in Figure 11.

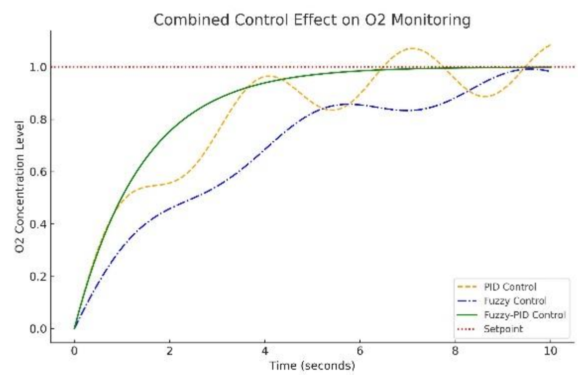


Figure 11. Comparative Experimental Control Effect Curve of the Three Algorithms.

The Fuzzy-PID algorithm addresses the shortcomings of both by combining the fast response of PID with the stability of Fuzzy control. It reaches 90% of the setpoint within 2.5 seconds and stabilizes with deviations of less than ±0.03 concentration units, outperforming both individual algorithms. This balance between speed and precision is essential in coal mine O2 monitoring, where timely and stable adjustments are critical to ensure safety and efficiency in dynamic and uncertain conditions.

VII. SYSTEM IMPLEMENTATION

The author used the theoretical framework of the coal mine gas intelligent IoT monitoring system based on the Fuzzy-PID algorithm proposed in this paper, which has been successfully developed and tested in a gas concentration monitoring and early warning project at a coal mine enterprise. The gas monitoring main screen of the system is shown in Figure 12. The system can monitor changes in gas concentration in real-time and can interlock and control multiple ventilation system variable frequency drives to dynamically adjust the gas concentration.

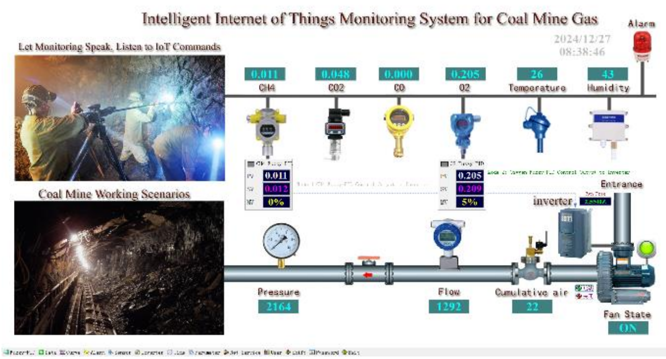


Figure 12. The Main Screen of the Intelligent IoT Monitoring System for Coal Mine Gas Using the Fuzzy-PID Algorithm.

The IoT control service software of the system was developed using the object-oriented integrated development tool Embarcadero RAD Studio XE, and the database system uses the lightweight and efficient SQLite. The mobile terminal part was developed using Android Studio to create the coal mine gas monitoring app. Considering the special needs for coal mine production safety, the mobile terminal only provides real-time browsing of operational parameters and data such as gas concentration, air volume, and air pressure, with no remote control operations allowed.

Since the system began its trial operation for more than half a year, it has shown significant advantages compared to traditional manual gas monitoring methods: it is more convenient to operate and has a higher degree of intelligence; the equipment operation alerts are accurate and the response is timely. This system provides strong technical support for intelligent gas concentration monitoring and safe production in coal mines, effectively reducing the risk of safety accidents and improving the level of automated management in coal mine operations.

VIII. CONCLUSIONS

The coal mine gas intelligent IoT monitoring system based on the Fuzzy-PID algorithm is an important component of smart mine construction. This system can collect and monitor various environmental parameters such as gas concentration, temperature, humidity, pressure, and air volume in real-time underground. Through IoT technology, it enables efficient data transmission and processing, ultimately achieving remote real-time control and early warning functions. By dynamically adjusting the operating status of the ventilation system (e.g., variable frequency drive fan control), the system can accurately control the underground gas concentration within a safe range, significantly improving the safety and intelligence of coal mine production.

In addition, the system uses the Fuzzy-PID algorithm, which can adaptively adjust control parameters based on changes in environmental conditions, improving control accuracy and response speed while reducing the need for manual intervention. At the same time, the IoT platform enables cross-regional collaborative monitoring and management of the coal mine, closely integrating gas monitoring with other mine operations such as ventilation, drainage, and more, further optimizing the overall efficiency and management level of the coal mine production process.

In summary, the coal mine gas intelligent IoT monitoring system based on the Fuzzy-PID algorithm has significant advantages in reducing underground operation risks, enhancing production safety, reducing labor intensity, and improving equipment reliability. It provides strong technical support for the intelligent production and safety management of coal mines.

ACKNOWLEDGMENT

The research is supported by the college student innovation training project. (Financing projects No. S202410702133).

REFERENCES

[1] L. Cen, "Research on key technologies of wireless gas monitoring system in underground coal mines," *Coal Mine Modernization*, vol. 34, no. 1, 2025, pp. 7–11.

[2] K. Du and Q. Li, "Research on characteristic analysis and early warning technology of gas monitoring values in coal mines," *Science and Technology Innovation and Application*, vol. 14, no. 32, 2024, pp. 94–98.

[3] B. Wang, "Daily management measures of coal mine gas monitoring and control system," *Mining Equipment*, no. 8, 2023, pp. 140–142.

[4] J. Zhang, "Application of ZigBee wireless communication technology in coal mine gas monitoring systems," *Mechanical Research and Application*, vol. 35, no. 4, 2022, pp. 181–183+186.

[5] Y. Shi, Y. Lian, L. Ju, et al., "Research and application of coal mine gas safety monitoring and control system," *Coal Technology*, vol. 42, no. 9, 2023, pp. 234–237.

[6] W. Liao, "Development of gas disaster early warning system in Jiangjiahe coal mine," *Coal Technology*, vol. 40, no. 11, 2021, pp. 128–131.

[7] X. Zheng, X. Tong, J. Guo, et al., "Research status and development trends of intelligent monitoring and early warning technology in coal mines," *Industry and Mining Automation*, vol. 46, no. 6, 2020, pp. 35–40.

[8] Z. Zhang and F. Feng, "Research on coal mine gas monitoring and early warning model based on SSA-SVM," *Ningxia Engineering Technology*, vol. 22, no. 4, 2023, pp. 359–365.

[9] C. Han, "Intelligent cloud platform for coal mine gas extraction system based on the Internet of Things," *Inner Mongolia Coal Economy*, no. 1, 2021, pp. 148–149.

[10] J. Dong, "Optimization design of intelligent early warning system for coal mine gas accidents," *Electromechanical Engineering Technology*, vol. 49, no. 7, 2020, pp. 249–250.

[11] J. Chen, Y. Huo, and Y. Xu, "Development of a dynamic sensor information control system for coal mines based on the Internet of Things," *Metal Mine*, no. 7, 2024, pp. 189–195.

[12] J. Zhang, "Design of collaborative control scheme for coal mine production equipment based on the Internet of Things," *Coal Technology*, vol. 41, no. 12, 2022, pp. 241–243.

[13] F. Wei, S. Wang, H. Deng, et al., "Research on the control of a single inverted pendulum based on fuzzy control algorithm," *Computer Simulation*, vol. 40, no. 3, 2023, pp. 320–325.

[14] H. Liu, Y. Lin, Z. Tian, et al., "Design of an intelligent air sieve system based on fuzzy control algorithm," *Food and Machinery*, vol. 39, no. 12, 2023, pp. 88–91.

[15] Q. Shen, C. Wan, X. Gao, et al., "Attitude control algorithm for high-speed spinning projectiles based on variable universe fuzzy control," *Journal of Beijing Institute of Technology*, vol. 42, no. 6, 2022, pp. 634–640.

[16] H. Liu, "Research on coal mine gas concentration monitoring system based on fuzzy intelligent control," *Coal Engineering*, vol. 51, no. 4, 2019, pp. 80–84.

[17] P. Gong, Y. Zhao, and M. Zhou, "Design of an intelligent mine ventilation control system based on fuzzy PID control," *Mining Machinery*, vol. 52, no. 12, 2024, pp. 58–62.

[18] Z. Zhang, Q. Dai, and Y. Zhu, "Gas blending concentration control based on improved fuzzy neural network PID," *Journal of Xi'an University of Science and Technology*, vol. 43, no. 2, 2023, pp. 388–397.

[19] Y. Xiao and X. Zhang, "Design of a gas variable frequency control system based on fuzzy control," *Automation and Instrumentation*, vol. 39, no. 7, 2024, pp. 32–36.

[20] Z. Zhang, Q. Dai, and Y. Zhu, "Gas blending concentration control based on improved fuzzy neural network PID," *Journal of Xi'an University of Science and Technology*, vol. 43, no. 2, 2023, pp. 388–397.

Study on Fuzzy Adaptive Based Synchronous Control of Dual Motor Deviation Coupling

Xingbo Wang
School of Armament Science and Technology
Xi'an Technological University
Xi'an, Shaanxi, China
E-mail: 571253768@qq.com

Baoji Ma
School of Armament Science and Technology
Xi'an Technological University
Xi'an, Shaanxi, China
E-mail: mabaoji@xatu.edu.cn

***Abstract*—To address the motor synchronization errors in the take-up system, an improved deviation-coupled synchronous control strategy based on fuzzy adaptive PID control is proposed. Traditional dual-motor synchronization methods struggle under conditions of high speed, highly dynamic operation, and significant disturbances, particularly in mitigating synchronization errors and system oscillations caused by parameter variations and sensor delays. This paper establishes a pendulum-angle feedback model, introduces a dynamic velocity compensation mechanism and adaptive synchronization gains, and develops a fuzzy adaptive PID controller. The controller dynamically adjusts PID parameters in real-time using pendulum-angle errors and their rate of change as inputs, achieving precise control of the motors' speed difference. Simulink simulation results indicate that, compared to traditional PID control, the proposed method reduces motor speed overshoot by 75% and decreases disturbance recovery time by over 50%. Experimental validations confirm that the method significantly reduces pendulum-angle fluctuations, enhances dynamic response speed, and improves system stability, thus fulfilling practical engineering requirements.**

Keywords-Take-Up System; Dual Motor Cooperative Control; Improved Deviation Coupling; Fuzzy Adaptive Control

I. INTRODUCTION

Dual motor synchronization control technology is widely used in textile processing, printing and packaging, cable production and paper and other fields, in which the winding take-up system is particularly stringent synchronization control accuracy requirements [1]. The precision of motor synchronization control is directly related to the stability of the winding tension, the neatness of the winding product and the quality of the final product. In the actual production process, due to load disturbance, system parameter changes and mechanical transmission errors and other factors, the traditional control method is difficult to ensure that the traction motor and take-up motor synchronization, which leads to tension fluctuations, product quality degradation, and even cause equipment failure [2].

Proportional-integral-derivative (PID) control, master-slave control and other methods commonly used in the industry, although the structure is simple, easy to realize the advantages, but in the face of complex working conditions, nonlinear disturbances and high-speed dynamic changes in the face of the obvious shortcomings [3-5]. Master-slave control depends on the rigid speed relationship, the lack of flexible adaptive adjustment ability, while the traditional PID control is overly dependent on the tension sensor feedback signal, the sensor itself exists in the precision error and signal lag further aggravate the system oscillation and control instability [6]. Therefore, for the special working conditions of high-speed precision winding, there is an urgent need for an advanced synchronous control method that does not need to directly rely on the feedback from the tension sensor, and at the same time has both fast dynamic response and robustness.

In order to solve the above problems, this paper proposes a fuzzy adaptive PID-based synchronous control strategy for dual-motor deviation coupling. By introducing a real-time feedback mechanism of the pendulum angle, and taking the pendulum deflection angle and its rate of change as inputs, a fuzzy adaptive algorithm is used to dynamically adjust the PID parameters, correct the speed difference between the motors in real time, and realize the fast and accurate compensation of the system synchronization error. This method effectively improves the dynamic response characteristics of the winding and take-up system, improves the anti-interference ability and synchronization control accuracy of the system, and meets the strict requirements of constant tension control in actual production.

II. CONSTANT TENSION TAKE-UP SYSTEM

The take-up system studied in this paper is mainly composed of traction motor, take-up motor and gravity adjustment pendulum, the structure sketch shown in Figure 1. Among them, the traction motor is responsible for pulling the wire forward at a constant speed, the take-up motor ensures that the wire is stably and neatly wound onto the reel by controlling the rotational speed, and the gravity pendulum is used to set and adjust the initial tension of the take-up.

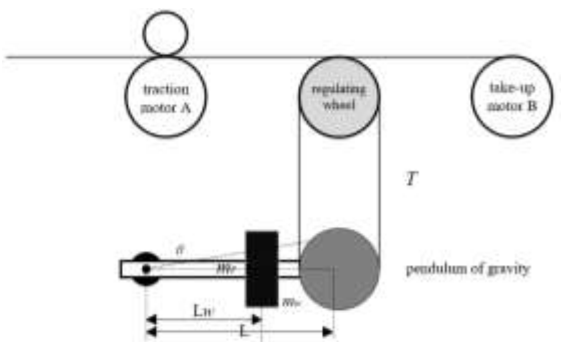


Figure 1. Schematic Diagram of the Mechanism

According to the sketch it can be seen that the gravitational moment generated by the traveler code and the gravitational moment of the pendulum itself can be expressed as:

$$M_w = m_w \cdot g \cdot L_w \cdot \cos\theta \tag{1}$$

$$M_r = m_r \cdot g \cdot \frac{L}{2} \cdot \cos\theta \tag{2}$$

where m_w is the mass of the vernier, m_r is the mass of the pendulum, L is the total length of the pendulum, L_w is the distance from the vernier to the center of rotation, θ is the deflection angle of the pendulum (measured from the horizontal position); and g is the acceleration of gravity.

The moment of inertia is satisfied when angular acceleration exists in the system:

$$M_I = J \cdot \alpha \tag{3}$$

where the pendulum rotational inertia is modeled as a thin rod rotating around an endpoint:

$$J = \frac{1}{3} m_r L^2 \tag{4}$$

where J is the rotational inertia of the pendulum and α is the angular acceleration.

According to the principle of moment equilibrium, the system satisfies:

$$2T \cdot L = M_w + M_r + M_I \tag{5}$$

The collapsing simplification yields the expression for the tension T as:

$$T = \frac{1}{2L}\left[g\left(L_w m_w + \frac{1}{2} L m_r \right)\cos\theta + \frac{1}{3} m_r L^2 \alpha \right] \tag{6}$$

It can be seen that the size of the tension is related to the position of the traveler code and geometric parameters, so by adjusting the position of the traveler code L_w can change the size of the tension, so as to realize the accurate setting of the initial tension. When the system is in the process of dynamic change $(\theta \neq 0, \alpha \neq 0)$, the pendulum deviates from the horizontal position offset angle and angular acceleration will generate dynamic errors in tension, resulting in the system's nonlinear and time-varying characteristics. It can be seen that the system lacks adaptive adjustment ability in the structural design, and the traditional motor synchronization control strategy is difficult to cope with the rapid change of the error [7].

III. SYSTEM COMPOSITION AND CONTROL FRAMEWORK DESIGN

In order to solve the above problems and make the control system adaptive, this paper proposes a fuzzy adaptive dual-motor deviation coupling synchronous control system based on fuzzy adaptive. Through the real-time feedback of the deflection angle of the back end of the pendulum, and the mathematical model of the pendulum deflection angle and the motor speed difference is constructed. Combined with the fuzzy adaptive control generates a new adjusted speed difference to restore the pendulum to the horizontal position. The fuzzy adaptive PID controller combines fuzzy rules and adaptive adjustment, which can effectively deal with the nonlinear, time-varying and strong coupling characteristics of the system, improve the dynamic response capability and robustness of the system, and ensure the stable and efficient operation of the system in complex environments.

A. Motor synchronization control strategy

The core of this control system is to realize efficient synchronization of traction motor and take-up motor through deviation coupling control strategy to ensure constant tension; the two motors can respond to the speed difference independently and recover synchronization quickly, reflecting the independence and coupling characteristics between motors.

Deviation coupling control is suitable for multi-motor control scenarios. The method through the speed compensator, according to the motor speed difference and rotational inertia ratio to calculate the compensation value, to make up for the rotational inertia difference caused by the speed error, so as to improve the synchronization performance of the system [8]. Figure 2 shows the structure of the deviation coupling control speed compensator.

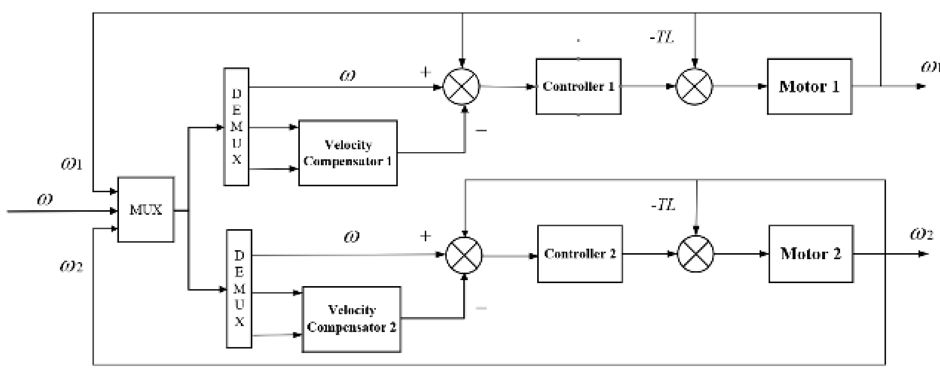


Figure 2. Deviation Coupling Control Structure Diagram

B. Improvement of deviation coupling control

In order to improve the accuracy of motor synchronization control, this paper introduces a real-time radius proportion adjustment mechanism on the basis of the deviation coupling control structure, which dynamically corrects the speed proportion of the two motors according to the actual radius of the take-up and pay-off motors to achieve the synchronization of line speed [9]. At the same time, for the traditional deviation coupling method using a fixed compensation value is difficult to adapt to the nonlinear characteristics of the system, easy to cause overshooting and oscillation, a synchronization gain parameter Kw based on the dynamic adjustment of the speed error is proposed. when the speed error is large, the Kw is automatically increased to quickly reduce the error; when the error is small, the Kw is reduced to avoid system oscillation.

In order to further improve the precision of winding tension control, a pendulum angle compensation module is added, and a fuzzy PID controller is utilized to generate a speed difference adjustment signal in real time according to the pendulum angle error (E) and the error change rate

(EC). The deviation coupling control structure after the above improvement is shown in Fig. 3, and the integrated radius proportional compensation, dynamic gain adjustment and pendulum angle feedback module realize the significant improvement of the motor synchronous control performance.

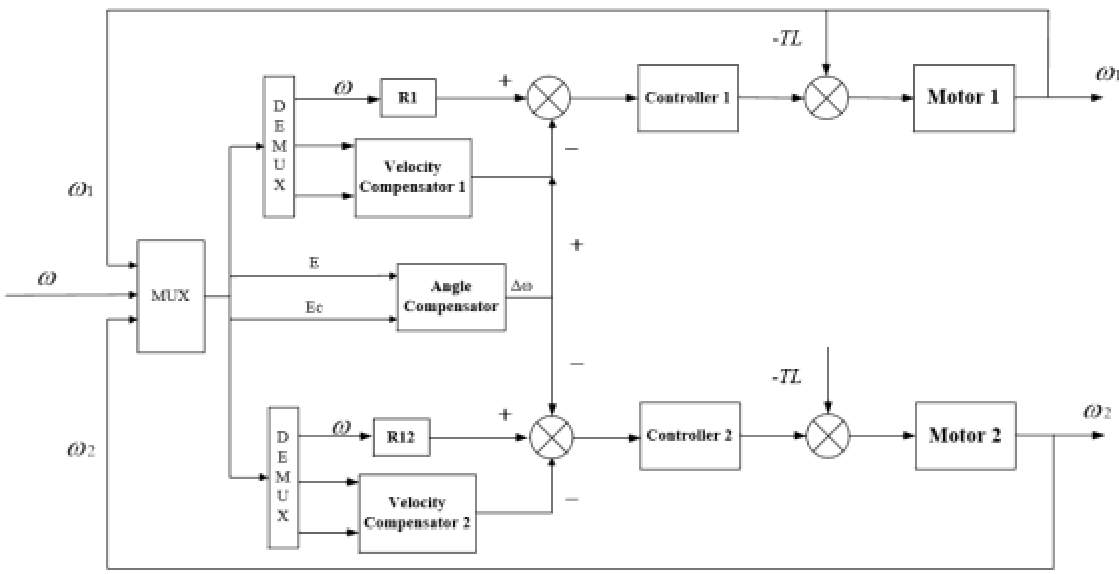


Figure 3. Improved Deviation Coupling Structure Diagram

C. Fuzzy Adaptive PID Controller Design

1) Principle of fuzzy adaptive PID controller

The fuzzy adaptive PID controller designed in this paper takes the desired angle of the pendulum as the target, and outputs the speed difference between the two motors for feedback regulation by monitoring the pendulum angle change in real time. The controller selects the pendulum angle error (E) and the error change rate (EC) as fuzzy input variables, and the PID parameter corrections (ΔK_p, ΔK_i, ΔK_d) as output variables [10]. The fuzzy inference rules are utilized to correct the PID parameters in real time to adapt to the dynamic demand and disturbance changes in the system operation, and the structure is shown in Fig. 4.

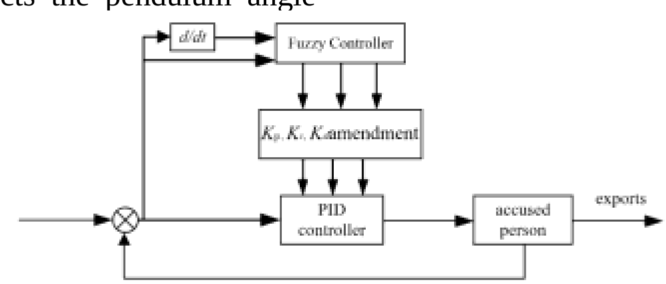


Figure 4. Fuzzy Adaptive PID Controller Structure Diagram

2) Fuzzy variable domain affiliation function

The domains of the fuzzy control input variables (E, EC) and output variables (ΔK_p, ΔK_i, ΔK_d) are divided into seven fuzzy sets {negative large (NB), negative medium (NM), negative small (NS), zero (Z), positive small (PS), positive medium (PM), positive large (PB)} in order to refine the description of the system error and its dynamic trend.

In order to enhance the anti-disturbance performance of the controller, the affiliation function adopts a triangular function, and the interval of the affiliation function of the input variables close to the center region (NS, Z, PS) is appropriately narrowed, so as to avoid frequent adjustment of the PID parameters. The specific affiliation function curve is shown in Fig.5.

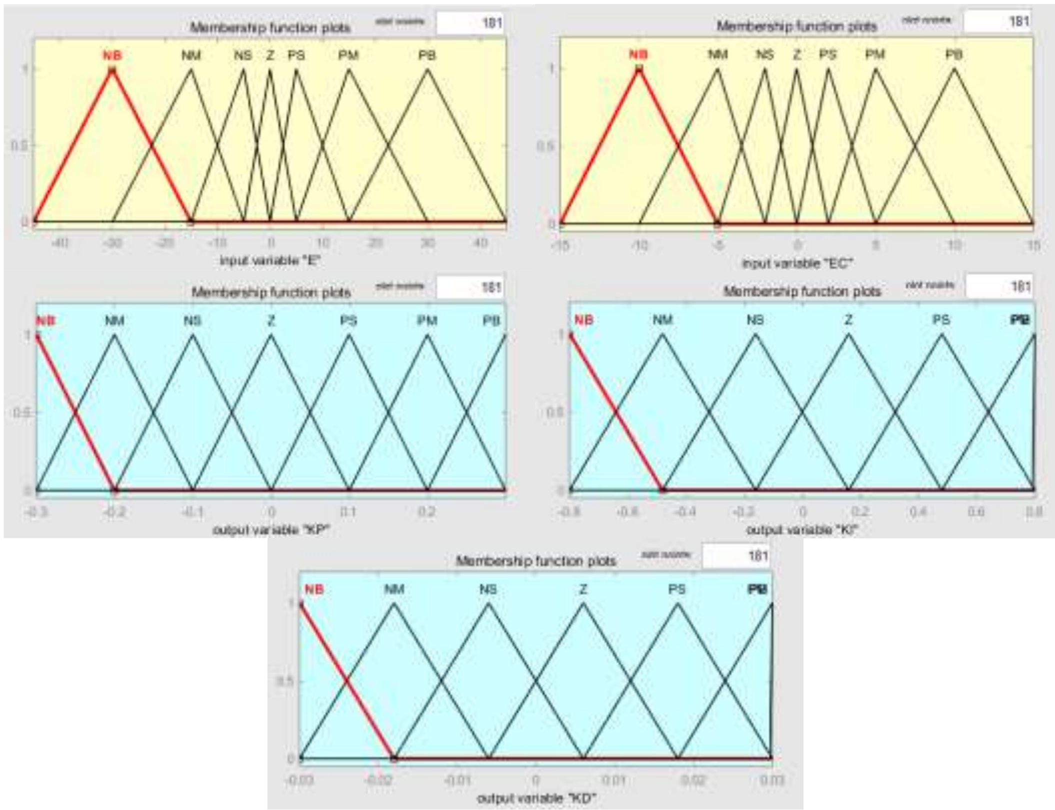


Figure 5. E, EC, ΔK_p, ΔK_i, ΔK_d Membership Functions

The fuzzy rules determine the adjustment strategies of output variables ΔK_p, ΔK_i, ΔK_d through different combinations of input variables E and EC to realize the real-time correction of PID parameters and adapt to the system dynamic control needs. The fuzzy rule control table is shown in Table 1.

TABLE I. FUZZY CONTROL TABLE FOR K_P, K_I AND K_D

E/EC	***NB***	***NM***	***NS***	***Z***	***PS***	***PM***	***PB***
NB	*PB/NB/PS*	*PB/NB/PS*	*PM/NB/PS*	*PM/NM/NB*	*PM/NS/NM*	*Z/NS/Z*	*Z/Z/Z*
NM	*PB/NB/NS*	*PM/NB/NS*	*PM/NM/NS*	*PM/NS/Z*	*Z/NS/Z*	*Z/Z/Z*	*Z/Z/Z*
NS	*PM/NM/Z*	*PM/NM/Z*	*PM/NM/NS*	*Z/NS/Z*	*Z/Z/Z*	*NM/PM/Z*	*NM/PB/PS*
Z	*PM/NM/Z*	*PM/NS/Z*	*Z/Z/Z*	*Z/Z/Z*	*Z/Z/Z*	*NM/PM/Z*	*NM/PB/Z*
PS	*PS/Z/Z*	*PS/Z/Z*	*Z/Z/Z*	*Z/Z/Z*	*NM/PM/Z*	*NB/PB/PS*	*NB/PB/Z*
PM	*Z/Z/PB*	*Z/Z/PB*	*NM/PS/Z*	*NM/PS/Z*	*NB/PB/Z*	*NB/PB/PB*	*NB/PB/PB*
PB	*PB/NB/PS*	*PB/NB/PS*	*PM/NB/PS*	*PM/NM/NB*	*PM/NS/NM*	*Z/NS/Z*	*Z/Z/Z*

IV. SIMULATION AND ANALYSIS

A. Simulation model

The simulation model of the dual motor synchronous control system is designed by combining the improved deviation coupling control with the SVPWM speed control strategy. In the Simulink simulation environment, the simulation model of dual motor synchronous control is built, and the overall simulation model is shown in Figure 6

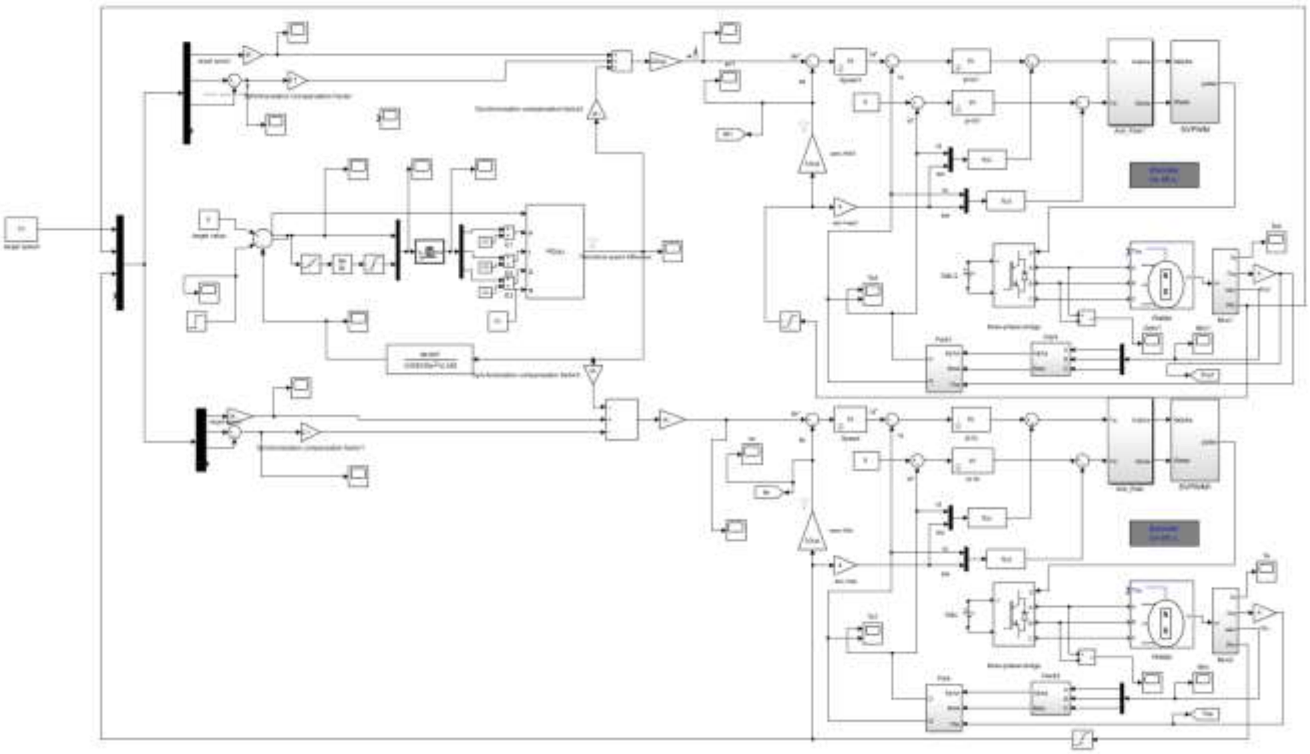


Figure 6. Two-Motor Synchronization Control Simulation Model

B. Simulation Analysis

In the simulation, the reference linear velocity of the two motors is set to 5 m/s, and the simulation time is 0.5 seconds. In the initial state, the radius ratio of the traction wheel and take-up wheel is 17:15, and the speed tracking curve of the two motors is shown in Fig. 7.

It can be seen that the speed following of the dual motors is good and the speed is up to the requirement. Subsequently the simulation compares the motor speed adjustment under conventional PID coupled control and fuzzy adaptive control after the pendulum angle deflection at the moment of 0.08 seconds. As shown in Fig.8-10.

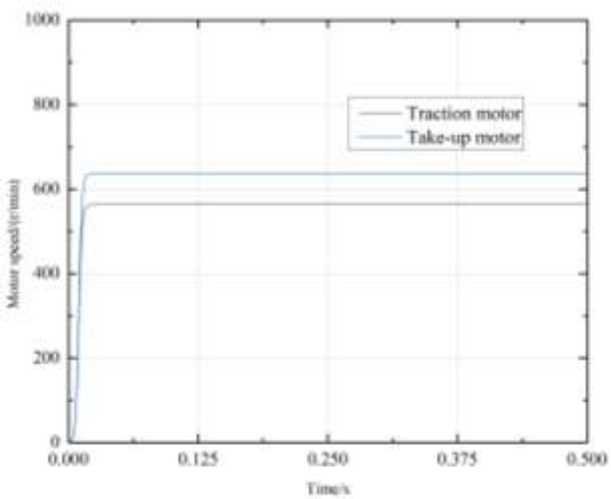


Figure 7. Two-Motor Speed Tracking Curve

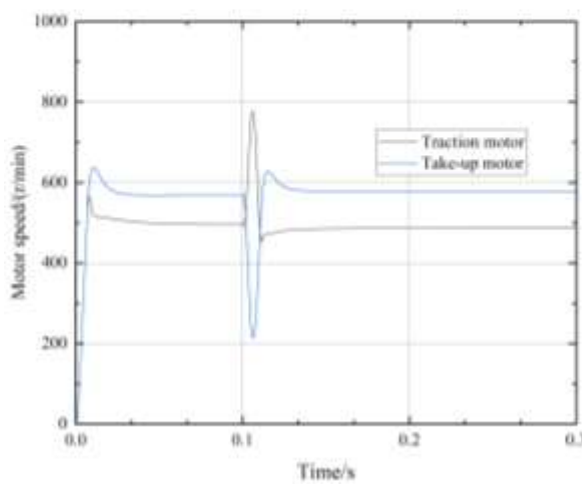


Figure 8. Conventional PID Coupling Control

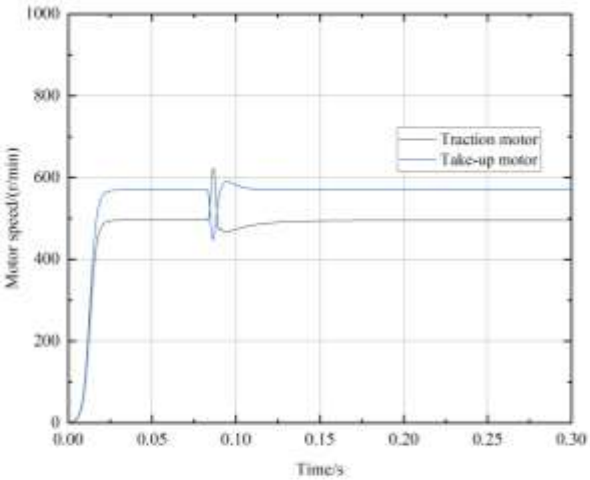


Figure 9. Fuzzy Adaptive PID Control

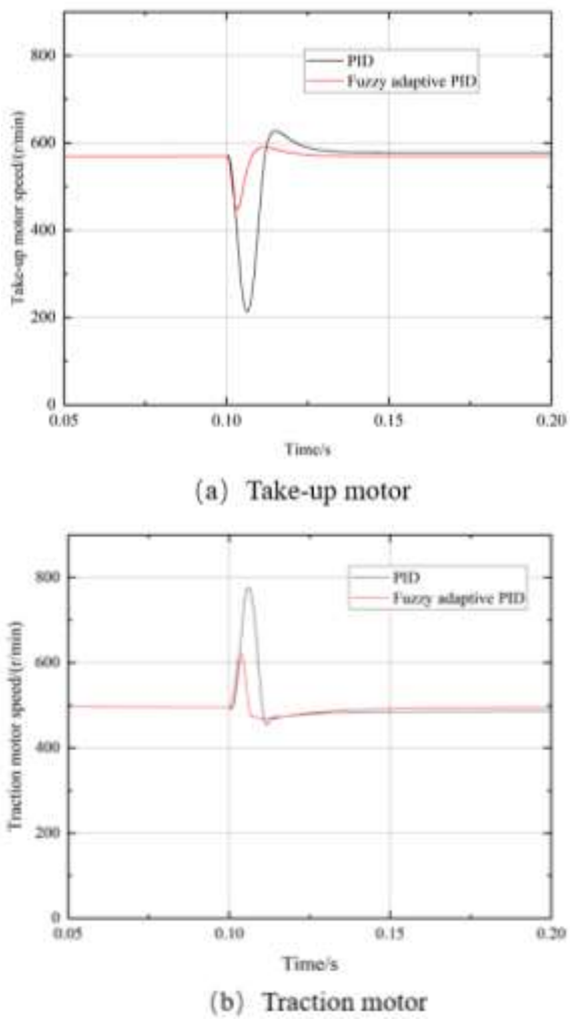


Figure 10. Motor Tracking Curves under Different Compensators

After the external disturbance occurs, the motor speed fluctuates drastically under the traditional PID control strategy, and the recovery stabilization process lasts about 0.08 s. The speed overshoot of the traction motor is as high as about 33%. The fuzzy adaptive PID control strategy can effectively suppress the disturbance, the system stabilization recovery time is shortened to less than 0.04 s, and the speed overshooting amplitude is significantly reduced to 8.3%, which improves the recovery speed by more than 50% and reduces the overshooting amplitude by about 75% compared with the traditional PID control. Simulation results show that the proposed fuzzy adaptive PID control strategy significantly improves the dynamic response performance and disturbance resistance of the winding system, and has better stability and practical engineering application value.

V. Experimental Validation

A. Hardware system design

In order to verify the practical effect of the above control scheme, this paper builds a dual-motor take-up system, as shown in Figure 11. The system uses Huichuan Easy PLC as the controller, in which the wire-rowing motor is MS1H1-75B30CB servo motor, and the take-up motor is MS1H1-40B30CB servo motor. Through the multi-channel Ether CAT communication module, to realize the communication between the servo motors, in order to real-time monitoring of the pendulum angle changes [11,12], the system selected a single-turn absolute angular displacement encoder BRT50 sensor.

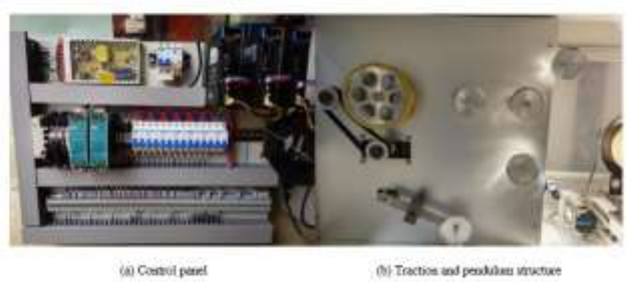


Figure 11. Constant Tension Fiber Collection System

B. Software system design

The software system in this paper adopts AutoShop software as the development environment [13], which is combined with

Simulink PLC Coder tool to realize the program design of fuzzy adaptive PID control strategy and improved deviation coupling control. The main program is responsible for the overall control logic, while the fuzzy adaptive PID controller realizes the dynamic correction of parameters and interacts with the servo drive in real time through the EtherCAT communication protocol, in order to ensure the precision and coordination of the control system, and the overall implementation flow of the working process of the system is shown in Fig12.

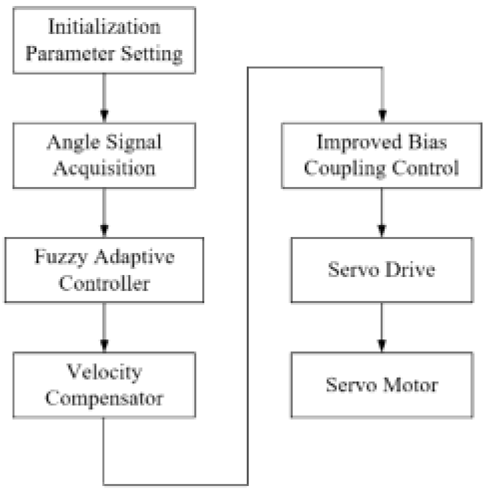


Figure 12. Working Process

In the above design, the PLC collects the pendulum angle signal in real time and dynamically adjusts the PID parameters according to the angle error (E) and the error change rate (EC) through the fuzzy adaptive PID controller to optimize the system response performance. At the same time, the synchronization gain parameter (Kw) is introduced into the improved deviation coupling control structure to calculate the speed difference between the motors in real time and dynamically adjust the synchronization ratio of the rotational speed based on the radius of the motors (R1, R2) in order to improve the adaptive ability of the system and avoid the oscillation and overshooting problems existing in the traditional deviation coupling control.

C. Experimental results and analysis

During the experiment, the speed monitoring curve of the two motors is shown in Figure 13. It can be seen that when the external disturbance occurs, the synchronization of the two motors is good, and the control system can respond quickly and make adjustments to ensure the smooth collection.

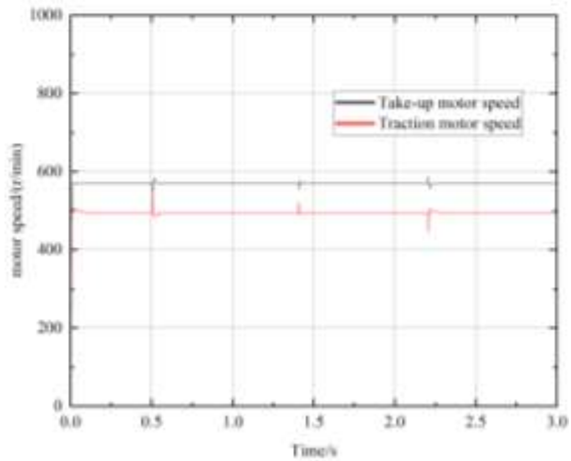


Figure 13. Real-Time Speed Curve

The adjustment ability of the traditional PID control and the fuzzy adaptive PID control is compared by recording the pendulum angle during the same 30s time period, as shown in Fig. 14.

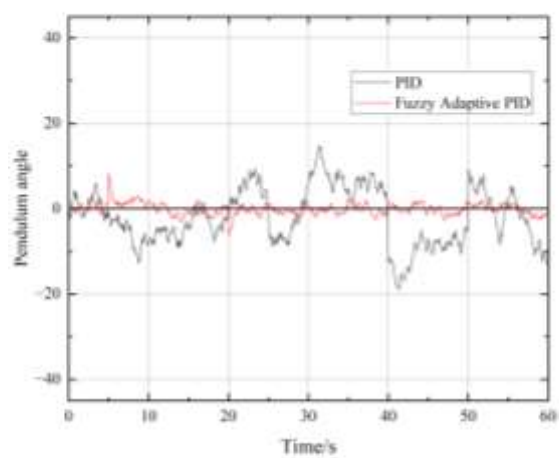


Figure 14. Pendulum Angle Variation

According to the experimental data graphs of the dynamic balance control of the pendulum, it can be seen that compared with the traditional PID control, the fuzzy adaptive PID control strategy significantly improves the stability and response performance of the pendulum. The fluctuation of the pendulum angle is about ±20 ° in the traditional PID control, while the fuzzy adaptive PID control can effectively limit the fluctuation of the pendulum to less than ±10°, which reduces the fluctuation amplitude by about 50%, and the system can be restored to the stable state faster, which reflects a better dynamic response capability and anti-interference performance.

Experimental verification shows that the proposed control scheme can not only effectively control the take-up tension, but also ensure the stable operation of the system under complex working conditions. The real-time data feedback and adaptive adjustment capability of the system make it perform well in dynamic adjustment, which greatly improves the reliability and efficiency of the collection system.

VI. CONCLUSION

In this paper, for the problem of insufficient synchronization control accuracy and weak disturbance resistance of dual motor in the winding and take-up system, an improved deviation coupling synchronization control strategy based on fuzzy adaptive PID is proposed. By constructing a real-time feedback model of the pendulum angle and introducing a dynamic adjustment mechanism of speed error compensation and synchronization gain parameters, the system realizes real-time accurate compensation of motor synchronization error. The simulation results show that compared with the traditional PID control method, the fuzzy adaptive PID strategy can effectively reduce the overshooting amplitude of the motor speed, significantly improve the recovery speed after perturbation, and improve the dynamic response performance of the system significantly. The experimental results further verify the effectiveness and practicability of the proposed method: the amplitude of pendulum angle fluctuation is reduced by about 50% compared with the traditional PID control, and the time for the system to recover from stability is significantly shortened, which shows good dynamic characteristics and anti-disturbance ability.

In summary, the fuzzy adaptive PID improved deviation-coupled synchronous control strategy proposed in this paper has obvious advantages in ensuring the constant winding tension, improving the winding quality and production efficiency, and can effectively cope with the parameter changes and complex working condition perturbations in the actual production, which has a good value of engineering applications and prospects for promotion.

REFERENCES

[1] X. D. Shang, Y. L. Liu, L. T. Ma, et al., "Research on coordinated control strategy for multi-motor systems based on fuzzy PID," Automation & Instrumentation, no. 3, pp. 13–17, 2021, doi: 10.14016/j.cnki.1001-9227.2021.03.013.

[2] Y. F. Lü, P. J. Zhang, W. Guo, et al., "Research on dual-motor synchronous control for rapid ammunition loading systems," Journal of Ordnance Equipment Engineering, vol. 44, no. 9, pp. 290–297, 2023.

[3] M. X. Chen, H. T. Zheng, J. C. Yin, et al., "Tension control of winding system based on fuzzy adaptive PID," Wool Textile Journal, vol. 49, no. 3, pp. 82–87, 2021.

[4] A. F. Hang and X. M. Wang, "Simulation of master-slave control for multi-motor parallel operation based on fuzzy PID," Computer Simulation, vol. 40, no. 6, pp. 316–320, 2023.

[5] W. Q. Zhu, D. W. Yuan, C. D. Cao, et al., "Design of constant tension winding system based on fuzzy adaptive PID," Industrial Control Computer, vol. 33, no. 7, pp. 25–27, 2020.

[6] L. Li, Multi-motor Synchronous Control in Optical Fiber Winding System [D], Taiyuan: North University of China, 2014.

[7] Y. M. Li, X. Miao, and F. J. Jiang, "Design of a multi-motor cross-coupled synchronous control system based on fuzzy PID controller," Electrical Engineering Technology, no. 6, pp. 118–120, 2019.

[8] H. Feng, Q. J. Du, D. X. Xu, et al., "Multi-motor synchronous predictive control based on mean deviation coupling," Instrument Technique and Sensor, no. 8, pp. 84–92, 2024.

[9] X. Y. Peng, W. Liu, and Q. Zhang, "Multi-motor synchronous control based on an improved deviation coupling structure," Journal of Hunan University (Natural Science), vol. 40, no. 11, pp. 77–83, 2013.

[10] C. L. Zhou, Research on Multi-PMSM Cooperative Control Based on Improved Deviation Coupling [D], Wuhu: Anhui Polytechnic University, 2022.

[11] S. J. Fu, Research and Development of Automated Cable Winding and Arranging Control System [D], Nanjing: Nanjing University of Aeronautics and Astronautics, 2020.

[12] S. L. Ren, Research on Precision Tension Control System in Fiber Winding Motion [D], Harbin: Harbin Institute of Technology, 2007.

C. W. Xu, Research on Rewinding Tension Control System of Gravure Printing Machine Based on Inovance Controller [D], Xi'an: Xi'an University of Technology, 2023.

Recurrent Neural Network-Aided BP Decoder Based on Bit-flipping for Polar Codes

Guiping Li
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: liguiping@xatu.edu.cn

Xiaojie Liu
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 2719130786@qq.com

Chang Yun
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: llxghp@sina.com

***Abstract*—Compared with SC decoding, BP decoding with the parallel mechanism has higher throughput and lower latency, which is more suitable for the demand of 5G scene. To further improve its FER performance and reduce the memory overhead, a recurrent neural network-aided bit-flipping BP decoding of polar codes is proposed. Firstly, it uses bit flip to correct the wrong decoded bits during the decoding iteration. And then, the offset min-sum approximation is used to replace multiplication operation. Lastly the improved recurrent neural network architecture is adopted to realize parameter sharing. The simulation shows that the proposed scheme has a better error correction ability with fewer flipping times, and can effectively reduce the computational resource consumption and extra memory overhead of BP decoding.**

Keywords-Polar Codes; Belief Propagation; Recurrent Neural Network; Bit-flip

I. INTRODUCTION

With the rapid development of Internet of Things (IoT), a growing number of physical devices are being connected at an unprecedented rate. The emerging 5G-IoT-centric applications like augmented reality, high-resolution video streaming, self-driven cars, smart environment and e-health care etc all require low latency, high throughput and ultra-reliable communication. Since polar codes proposed by E. Arikan in 2008 based on the polarization phenomenon of channel [1] are low complexity codes and can be proved to achieve Shannon capacity theoretically under successive cancellation (SC) decoding when the code length approaches infinity. Due to these excellent characteristics, polar codes are selected as the control channel coding in the enhanced Mobile Broadband (eMBB) scenario in the 5G mobile communication system plan. Since 2016, polar codes have been from theoretical research to engineering application less than 10 years after they were proposed.

In the recent academic research, the decoding algorithms of polar codes mainly fall into the following two categories, namely the Successive Cancellation (SC) [2] decoding and the Belief Propagation (BP) [3] decoding. Among them, SC decoding include its derivative decoding Successive Cancellation List (SCL) [4,5] and BP decoding are two kinds of commonly used algorithms in polar code decoding algorithm. The SC decoding algorithm has low computational complexity, but is limited by the nature of serial decoding, resulting in low throughput and high latency in the decoding process. Compared with the SC decoding algorithm, the BP decoding algorithm has higher throughput and lower decoding latency due to its parallel decoding characteristics, but it also leads to an increase in computational complexity, at the same time, the BP algorithm

DOI: 10.2478/ijanmc-2025-0004

iteratively the characteristics of decoding also make the decoding delay increase with the increase of the number of iterations.

In recent years, with the rapid development of deep learning, neural network technology has injected fresh blood into many fields. Due to the similarities between the neural network and the decoding network structure, many scholars have tried to apply the neural network to polar code decoding to reduce the decoding latency compared with the existing traditional decoding methods. Tobias Gruber [6] proposed a polar codes decoding scheme based on Deep Neural Network (DNN) in 2017, using DNN to learn the mapping relationship of information before and after decoding can effectively replace the traditional complex decoding algorithm at the receiving end, simulation proved that its decoding performance can reach the maximum a posteriori probability (MAP) performance close to that of the CA-SCL decoding algorithm. At present, the polar codes decoding algorithm based on neural network reduces the decoding latency to a certain extent, at the same time, the neural network decoding algorithm [7,8] still has certain limitations, the performance of the code has a great impact, and the memory overhead of the neural network BP decoding algorithm, the decoding error correction performance and the difficulty of learning the codeword structure mapping of SC decoding still need to be improved. Therefore, improving the generalization ability of neural network decoding to achieve optimal decoding performance with lower complexity is of great significance in future research.

Compared with BP decoding, SC decoding and the improved SC algorithm can achieve better channel capacity and lower block error rate (BLER), however, SC decoding has low throughput due to its sequential processing characteristics, while BP algorithm has outstanding performance in parallelization and low decoding latency. In recent years, the research of BP decoding is also trying to improve the decoding performance of BP while maintaining its advantages. One way to optimize the performance of BP decoding algorithm is to add neural network to assist the BP decoding process, expand the BP iterative decoding structure into the neural network architecture [9], and optimize the minimum and approximate scaling factors to compensate for the performance loss through deep learning. In reference [10,11], the BP decoding algorithm based on neural network reduces the total number of iterations and the overall complexity before convergence by scaling the messages with trainable weights, but it does not solve the problem of the lack of error correction performance of BP decoding. In order to solve a large number of multiplication operations caused by BP neural network decoding, and further reduce the block error rate of the BP decoding algorithm, this paper proposes a recurrent neural network-assisted polar codes bit-flipping (BF) BP decoding algorithm [12-15].

Firstly, this paper introduces some basic theories of polar codes and deep learning, then introduces the construction of bit-flipping and critical set of polar codes BP decoding algorithm. Then mainly introduces the polar codes bit flip BP decoding algorithm based on RNN and makes simulation analysis, and finally, a summary is made and the future of BP decoding is prospected.

II. KEY THEORIES AND ALGORITHMS

A. Polar Code Decoding Theory

Through the phenomenon of channel polarization, it is found that for any $N = 2^n$ independent channels, the channel capacity of some of the polarized channels tends to be polarized, besides, the coding complexity of polar codes is very low, which is mathematically a matrix multiplication operation. At the same time, the proposal of polar codes provides a factual basis for the proof of Shannon's noisy channel coding theorem, which further improves Shannon's theorem. Therefore, polar codes are considered to be an important breakthrough in the coding theorem, the process of polar codes channel polarization conversion is shown in Figure 1.

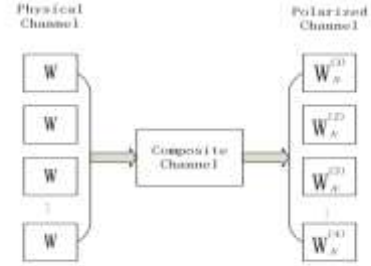


Figure 1. Polarization Transformation.

In the process of constructing the polar code of (N,K), first the information bit length of the source u is K, after polar coding, it becomes x, and the code word length is set to N. The K information bits and other frozen bits of $(N-K)$ will be arranged to the reliable bit channel and the unreliable bit channel respectively, and then multiplied by the coding matrix $\mathbf{G}_N$ of the polar code, which satisfies the formula:

$$\mathbf{x}^N = \mathbf{u}^N \mathbf{G}_N = \mathbf{u}^N \mathbf{F}^{\otimes n} \mathbf{B}_N \tag{1}$$

Where $\mathbf{B}_N$ is the bit-reversal permutation matrix, $n = \log_2 N$ and $\mathbf{F}^{\otimes n}$ represents the n-th Kronecker power of $\mathbf{F} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{1} \end{bmatrix}$.

B. BP Decoding Algorithm

SC decoding algorithm is a serial decoding algorithm, which has high latency and low throughput in the decoding process, in order to better solve the decoding latency problem of serial decoding, BP decoding is applied to polar codes decoding. The main process of BP decoding is to adaptively adjust the check matrix according to the reliable information in each iteration, and sparse it to obtain enhanced reliability information and decide the decoding, the BP factor diagram of a (N,K) polar codes has $N \times (n+1)$ nodes and $n = \log_2 N$ stages, the factor diagram of a $(8,4)$ polar codes is shown in Figure 2 and Figure 3.

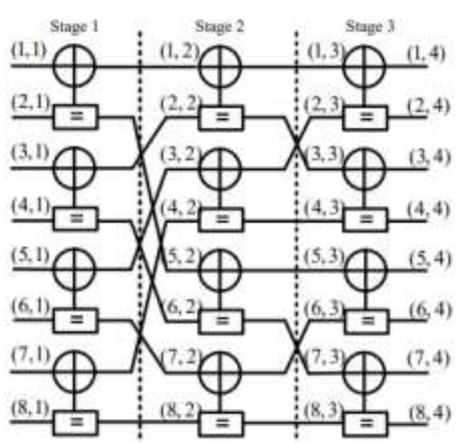


Figure 2. Factor diagram of BP decoding algorithm for $(8,4)$ polar codes.

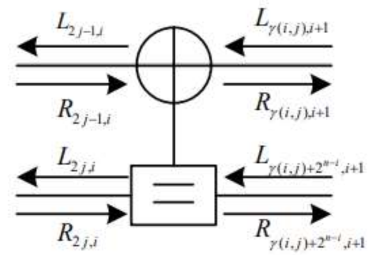


Figure 3. Unit factor diagram of BP decoding algorithm for polar codes.

When performing the BP iterative operation, each node (i, j) has two iterative calculation formulas from right to left and from left to right, by converting the information received by the receiver into log-likelihood ratio information, the for the iterative decoding operation of soft information. The information from left to right is expressed as $R_{i,j}^{(t)}$, and the information from right to left is expressed as $L_{i,j}^{(t)}$, the specific iterative rule for iteratively updating the likelihood ratio information is the following equation:

$$\begin{cases} L_{i,j}^{(t)} = g\left(L_{i+1,2j-1}^{(t)}, L_{i+1,2j}^{(t)} + R_{i,j+\frac{N}{2}}^{(t)}\right) \\ L_{i,j+\frac{N}{2}}^{(t)} = g\left(R_{i,j}^{(t)}, L_{i+1,2j-1}^{(t)}\right) + L_{i+1,2j}^{(t)} \\ R_{i+1,2j-1}^{(t)} = g\left(R_{i,j}^{(t)}, L_{i+1,2j}^{(t-1)} + R_{i,j+\frac{N}{2}}^{(t)}\right) \\ R_{i+1,2j}^{(t)} = g\left(R_{i,j}^{(t)}, L_{i+1,2j-1}^{(t-1)}\right) + R_{i,j+\frac{N}{2}}^{(t)} \end{cases} \tag{2}$$

First, the log-likelihood ratio information is passed from the rightmost to the leftmost, and then from the leftmost to the rightmost, to realize an iterative decoding operation, and the above-mentioned likelihood ratio information is updated in this process. Among them, $g(x,y) = \ln\frac{1+xy}{x+y}$, since the logarithmic calculation process is more complicated, the min-sum(MS) approximation [9] is generally used to simplify $g(x,y)$:

$$g(x,y) \approx sign(x) \times sign(y) min(|x|,|y|) \tag{3}$$

At the beginning of iteration, log-likelihood ratio information is initialized. The initial values $R_{1,j}^{(1)}$

and $L_{n+1,j}^{(1)}$ of right information log-likelihood ratio and left information log-likelihood ratio can be obtained by initializing log-likelihood ratio, which are expressed as follows:

$$R_{1,j}^{(1)} = \begin{cases} 0, & \text{if } j \in A \\ \infty, & \text{if } j \in A^{c} \end{cases} \tag{4}$$

$$L_{n+1,j}^{(1)} = \ln \frac{P(y_j | x_j = 0)}{P(y_j | x_j = 1)} \tag{5}$$

Since the BP algorithm is an iterative decoding algorithm, the number of iterations is involved, the superscript of the letter of the initialization formula above represents the number of iterations, because it is initialization, the superscript is 1, indicating the first iteration. Among them, A represents the set of information bits, and A^c represents the set of frozen bits, when the number of iterations reaches the set T times, the next step is the decision stage in the decoding process, the source data u_1^N is judged according to the likelihood ratio information of the leftmost node, and then the estimated value of the bit $\hat{u}_1^N$ is obtained, the specific judgment rules are as follows:

$$\hat{u}_j^N = \begin{cases} 0, & \text{if } L_{1,j}^{T} \geq 0 \\ 1, & \text{if } L_{1,j}^{T} < 0 \end{cases} \tag{6}$$

C. Deep Learning Theory

After artificial intelligence has achieved extensive applications and good results in various fields, the concept of deep learning is more and more familiar to people. The most common model of deep learning is deep neural network, as a popular learning model in the academic world, various academic fields try to bring deep learning into the original field, hoping to obtain better performance.

Each neuron of the deep neural network is not only connected with all the neurons in the previous layer to receive the information from them, but also connected with the neurons in the latter layer to transmit the current information. The number of neuron nodes in the input layer is affected by the data to be learned and needs to be consistent with the dimension of the learning data, connected to the input layer is the hidden layer, which receives the information from the input layer. A DNN model can be abstracted as a function f that maps the input $x_0 \in \mathbb{R}^{N_0}$ to the output $\mathbf{y} \in \mathbb{R}^{N_L}$:

$$\mathbf{y} = f(\mathbf{x}_0; \theta) \tag{7}$$

Convolutional Neural Network(CNN) is a deep neural network with a convolutional structure, the reason why the convolutional structure can offset the memory requirements as the number of network layer increases, three of which are crucial. First is the local receptive field, for a two-dimensional image, the connection between each other is usually only local, and it can be considered that the characteristics of each block will only affect the characteristics of the image around it, features are only affected by the features of the images connected to it, so low-level neurons learn local low-level features, and high-level connections are used to statistically synthesize each low-level feature, and gradually learn global features. The second is weight sharing, a convolution kernel is equivalent to a filter, which is used to extract features of one dimension, when the same feature appears in different dimensions, the previous convolution kernel is used to extract this feature, so that different dimensions a convolution kernel is shared, which not only reduces the parameters of the convolution kernel, but also allows the redundant convolution kernel to extract more features. The third is the pooling layer, pooling is a form of down sampling, there are many forms of nonlinear pooling functions, of which max pooling and average pooling are two common methods, through the pooling layer, the feature dimension can be reduced, so that learning fewer parameters is conducive to the establishment of the model, reducing the task of learning, and avoiding the occurrence of overfitting to a certain extent. Therefore, the convolutional neural network is a neural network algorithm that can effectively split the data.

Recurrent Neural Network(RNN) are also called feedback neural networks, the existence of feedback makes the recurrent neural network a nonlinear dynamic system that can be used to solve problems such as associative memory and optimization calculation, which is also the biggest difference between the recurrent neural network and the feedforward neural network. The recurrent neural network with fixed weights, external input and internal state can be divided into two network models: static field neural network model and local field neural network model. Among them, a typical well-known local field neural network model is called Hopfield-type neural network, and the optimization neural network mainly used to solve linear variational inequality and linear complement problem is one of the more famous static field neural network models.

As the excellent performance of deep neural network has been widely concerned by scholars from all walks of life, the application of neural network in the field of communication has gradually increased, the decoding process can be regarded as the process of signal classification, so the features contained in the coding structure can be learned by training the neural network. Some scholars have applied DNN to the decoding of linear codes and obtained better performance results, using the neural network decoder method, since the network training is completed offline, the decoding latency is not involved in the actual calculation. At present, the polar codes decoding algorithm based on neural network reduces the decoding latency to a certain extent, at the same time, the neural network decoding algorithm still has certain limitations, however, the memory overhead and decoding error correction performance of the neural network-based BP decoding algorithm still need further optimization. Therefore, improving the generalization ability of neural network decoding to achieve optimal decoding performance with lower complexity is of great significance in future research.

III. PRELIMINARY WORKS

A. *Bit-flip*

Incorrect decoding of information bits due to the BP decoding message passing algorithm can lead to error propagation, negatively affecting the reliability and accuracy of many other bits. Bit flipping is an auxiliary mechanism to the decoding process that guesses and flips potentially incorrect decoded bits before restarting the decoding process, therefore, precise bit flipping can effectively improve the BLER performance of polar codes. In order to solve the problem of error propagation, the BF mechanism flips the previously estimated value of u_j^N and sets the prior knowledge of $\hat{u}_j^N$ to infinity, then, the error-propagated decoding information is corrected by using infinite prior LLR instead of directly modifying the hard decision value of the flipped bit, therefore, the initialized value of R_0 in formula (4) is modified as:

$$R_{0,j}^{(1)} = \begin{cases} 0, & \text{if } j \in \{A \setminus F\} \\ \infty \times (2\hat{u}_j^N - 1), & \text{if } j \in F \\ +\infty, & \text{if } j \in A^c \end{cases} \tag{8}$$

where F is the set of flip positions, set to infinity.

B. *Critical set*

During the transmission of polar codes, the main reason for decoding errors is error propagation, by flipping the first error, the decoding performance can be improved. However, the biggest defect of the traditional flip set is that if the first wrong bit is not found, the performance will hardly be improved, and most of the time, there is more than one wrong bit, how to select unreliable information and build the correct flip set is the biggest difficulty. However, the search range of the first error bit is the whole unfrozen bit set, in order to narrow the search range of the first error unfrozen bit in decoding, the critical set (CS) composed of most error-prone bits was proposed and adopted in reference [8-14]. The critical set is constructed according to the structure of polar codes, in which the first node of all information sub-trees is high risk, only by selecting bits for BF from CS, fewer flip attempts can be achieved and lower latency can be achieved. In addition, the reference [12] puts forward the ω-order CS to flip ω-easy dislocations at the same time, at the cost of increasing 2^{ω}-times, with better error correction ability, in the literature [14,15], the BP decoding algorithm of polar codes based on recurrent neural network and the BP decoding algorithm of polar codes bit-flip based on convolutional neural network were proposed, which achieved good decoding and error correction ability, but have large memory overhead and flipping attempts.

$E\left\{L_N^{(i)}\right\}$ represents the expected value of log-likelihood ratio of the i_{th} polarized channel, then the error rate of the ith polarized channel can be expressed as

$$P_e(u_i) = Q\left(\sqrt{E\left\{L_N^{(i)}\right\}/2}\right) \tag{9}$$

Where $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^{+\infty} e^{-\frac{\alpha^2}{2}} \mathrm{d}\alpha$, selecting the indexes of k polarized channels with the largest $E\left\{L_N^{(i)}\right\}$ to form the information bit set A.

The polar codes structure can be represented by a complete binary tree, as shown in Figure 4 [12]. The black node indicates that all its leaf nodes are information bits, while the white node indicates that all its leaf nodes are frozen bits, and the gray node indicates that its leaf nodes contain both information and frozen bits, black nodes are nodes with coding rate of 1, and the critical set needs to take the index of the first bit in each node with coding rate of 1.

$$CS = \bigcup_i \mathcal{R}_i[1] \tag{10}$$

Where $\mathcal{R}_i$ represents the i_{th} node with a rate of 1, and $\mathcal{R}_i[1]$ represents the index of the first bit in $\mathcal{R}_i$.

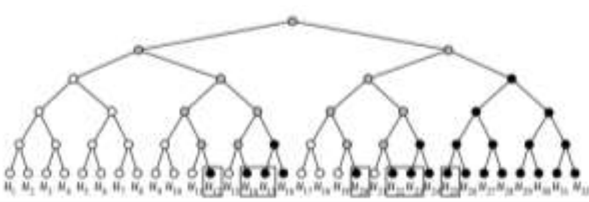


Figure 4. Binary Tree Structure Diagram(polar (32,16)).

The bit inversion of CS-1 is the same one-dimensional vector as CS, however, it contains a large number of error-prone exponents (about $|CS|^{\omega}$), which need to be tried by bit inversion decoding. In order to avoid the bit-flip decoding index attempt and obtain lower decoding latency, the truncated version of CS-ω was proposed in [12], CS-ω is a $|CS|\times\omega$ matrix, and one row of CS-ω represents the ω index of ω flip bits in one BP decoding. Each line of CS-ω is shown as follows, first of all, according to the error rate $P_e(u_i)$ and $i \in CS$, the elements in CS are sorted in descending order:

$$\begin{aligned} &\left\{\mathrm{CS}(k_1), \mathrm{CS}(k_2), \ldots, \mathrm{CS}\left(k_{|CS|}\right)\right\} \\ &\text{s.t. } P_e\left(u_{\mathrm{CS}(k_i)}\right) \geq P_e\left(u_{\mathrm{CS}(k_{i+1})}\right) \end{aligned} \tag{11}$$

Next, the i_{th} line in $CS-\omega$ is expressed as:

$$\begin{aligned} &CS-\omega(i) = (j_1, j_2, \ldots, j_\omega), 1 \leq i \leq |CS| \\ &j_1 = CS(k_i) \\ &j_2, j_3, \ldots, j_\omega \in \mathcal{A} \\ &j_1 < j_2 < \cdots < j_\omega \end{aligned} \tag{12}$$

Based on the above description of the binary tree structure of polar codes, reference [12] constructs the critical set CS according to formula (10) and the binary tree structure of polar codes, for more detailed information about bit flip and critical set, please refer to reference [8-12].

Neural network BP decoding algorithm

The min-sum approximation function avoids the complicated logarithmic operation, simplifies the calculation process of the algorithm, and uses the early termination criterion to reduce the redundant iterative process and the decoding time of BP algorithm. However, the above improvements still can't change the characteristics that BP algorithm is an iterative decoding algorithm, and can't achieve considerable decoding performance under the limited number of iterations.

In recent years, due to the excellent performance of deep neural network, which has been widely concerned by scholars from all walks of life. The application of neural network in the field of communication has gradually increased, the decoding process can be regarded as the process of signal classification, so the features contained in the coding structure can be learned by training the neural network. Some scholars have applied DNN to the decoding of linear codes and obtained better performance results. Using the neural network decoder method, since the network training is completed offline, the decoding latency is not involved in the actual calculation. This neural network decoding model has a highly parallel structure, so it is also called a one-shot decoding process, in addition, with the emergence of a large number of open-source neural network frameworks, such as Tensorflow, Pytorch, etc, deep learning can easily carry out high-speed calculation.

In the literature [7], the author first proposed a DNN-based polar codes BP decoding algorithm (DNN-BP), the learnable parameters are set in, so as to make up for the performance loss caused by the minimum sum approximation. The specific iterative rule formula (2) for updating the iterative likelihood ratio information is updated to the following formula:

$$\begin{cases} L_{i,j}^{(t)} = \alpha_{i,j}^{(t)} \cdot g'\left(L_{i+1,2j-1}^{(t)}, L_{i+1,2j}^{(t)} + R_{i,j+N/2}^{(t)}\right) \\ L_{i,j+N/2}^{(t)} = \alpha_{i,j+N/2}^{(t)} \cdot g'\left(R_{i,j}^{(t)}, L_{i+1,2j-1}^{(t)}\right) + L_{i+1,2j}^{(t)} \\ R_{i+1,2j-1}^{(t)} = \beta_{i+1,2j-1}^{(t)} \cdot g'\left(R_{i,j}^{(t)}, L_{i+1,2j}^{(t-1)} + R_{i,j+N/2}^{(t)}\right) \\ R_{i+1,2j}^{(t)} = \beta_{i+1,2j}^{(t)} \cdot g'\left(R_{i,j}^{(t)}, L_{i+1,2j-1}^{(t-1)}\right) + R_{i,j+N/2}^{(t)} \end{cases} \tag{13}$$

Where $\alpha_{i,j}^{(t)}$ and $\beta_{i,j}^{(t)}$ respectively represent the scaling factors of the log-likelihood ratio of left information transmitted from right to left and the log-likelihood ratio of right information transmitted from left to right by nodes (i, j) in the t iteration of polar codes BP decoding.

Although DNN-BP can better solve the approximate loss problem of min-sum, there are still two problems in this deep neural network architecture. First of all, the scaling factor set by this algorithm to reduce the approximate loss brings a lot of multiplication operations to the algorithm, which leads to a significant increase in the computational complexity of BP decoding algorithm based on deep neural network. In addition, the storage space is occupied by the storage of a large number of scaling factors in the deep neural network architecture. Although the BP decoding algorithm based on DNN reduces the total number of iterations before the convergence of the decoding algorithm by scaling the messages with trainable weights, it does not solve the problem of error propagation in the BP decoding algorithm. The incorrect decoding of information bits in the BP decoding message transmission algorithm may lead to error propagation, which will have a negative impact on the reliability and accuracy of many other bits, in order to further solve the problems existing in the BP decoding algorithm of polar codes based on DNN and reduce the BLER of BP decoding algorithm, this paper proposes a BP decoding algorithm of polar code bit-flip based on recurrent neural network.

IV. BP DECODER OF POLAR CODES BIT-FLIP BASED ON RECURRENT NEURAL NETWORK

There is a big difference between deep neural network and recurrent neural network, DNN cannot model changes in time series. In an ordinary fully connected network or CNN, the processing is independent at each moment, while in RNN, the output of neurons can directly act on itself in the next time period, the input of neurons in the i_{th} layer at m time, except the output of neurons in the $(i-1)$ layer at that time, the output of the neuron can directly act on itself in the next time period. Therefore, the RNN-based BP decoding algorithm will reuse the weights in different iterations, thereby realizing parameter sharing among multiple iterations of the neural network, inspired by the recurrent neural network architecture in literature [15], a BP decoding architecture of recurrent neural network based on minimum and approximate scaling offset is proposed. Therefore, the specific iterative rule formula (13) for the previous iteration to update the log-likelihood ratio information is replaced by the following formula:

$$\begin{cases} L_{i,j}^{(t)} = \alpha_{i,j} \cdot g\left(L_{i+1,j}^{(t-1)}, L_{i+1,j+N/2^i}^{(t-1)} + R_{i,j+N/2^i}^{(t)}\right) \\ L_{i,j+N/2^i}^{(t)} = \alpha_{i,j+N/2^i} \cdot g\left(R_{i,j}^{(t)}, L_{i+1,j}^{(t-1)}\right) + L_{i+1,j+N/2}^{(t-1)} \\ R_{i+1,j}^{(t)} = \beta_{i+1,j} \cdot g\left(R_{i,j}^{(t)}, L_{i+1,j+N/2^i}^{(t-1)} + R_{i,j+N/2^i}^{(t)}\right), \\ R_{i+1.i+N/2^i}^{(t)} = \beta_{i+1.i+N/2^i}^{(t)} \cdot g\left(R_{i,j}^{(t)}, L_{i+1,j}^{(t-1)}\right) + R_{i.i+N/2^i}^{(t)} \end{cases} \tag{14}$$

The network architecture of BP decoding

algorithm based on RNN is shown in Figure 5 and Figure 6, in order to reduce the large number of multiplication operations caused by the minimum sum approximation, the Offset Min-Sum (OMS) approximation proposed in the literature [13] is used to improve the BP decoding algorithm, and the $g(x,y)$ function is updated as the following formula:

$$g(x,y) \approx \operatorname{sgn}(x)\operatorname{sgn}(y) \times \max\left(\min(|x|,|y|) - \beta, 0\right) \tag{15}$$

Compared with the scaling factor minimum and approximation algorithm, the minimum and approximation algorithm is different in that a linear rectification function $f_{\mathrm{ReLU}}(x)$ and a learnable offset β are added, which play the roles of nonlinear activation and substitution of multiplication respectively, this function is designed to provide better nonlinear fitting ability for neural networks, at the same time, similar to the scaling factor minimum sum algorithm. In order to improve the system performance, different offsets are applied to different nodes in the factor graph, the specific iterative rule formula (14) for iteratively updating the log-likelihood ratio information is further updated as:

$$\begin{cases} L^{(t)}_{(i,j)} = g\left(L^{(t-1)}_{i+1,j}, L^{(t-1)}_{i+1,j+\frac{N}{2^i}} + R^{(t)}_{i,j+\frac{N}{2^i}}, \beta_{L^t_{i,j}} \right) \\ L^{(t)}_{i,j+\frac{N}{2^i}} = g\left(R^{(t)}_{i,j}, L^{(t-1)}_{i+1,j}, \beta_{L^{(t)}_{i,j+\frac{N}{2^t}}} \right) + L^{(t-1)}_{i+1,j+\frac{N}{2}} \\ R^{(t)}_{i,j} = g\left(R^{(t)}_{i,j}, L^{(t-1)}_{i+1,j+\frac{N}{2^i}} + R^{(t)}_{i,j+\frac{N}{2^i}}, \beta_{R^t_{i+1,j}} \right) \\ R^{(t)}_{i,j+\frac{N}{2^i}} = g\left(R^{(t)}_{i,j}, L^{(t-1)}_{i+1,j}, \beta_{R^t_{i+1,j+\frac{N}{2^i}}} \right) + R^{(t-1)}_{i,j+\frac{N}{2^i}} \end{cases} \tag{16}$$

Where $\beta_{L^t_{i,j}}$ and $\beta_{R^t_{i,j}}$ represent the offset of the left information $L^{(t)}_{(i,j)}$ and the right information $R^{(t)}_{i,j}$ in the i_{th} iteration process, respectively added during calculation. When the offsets are all set to 0, the algorithm is simplified to a common min-sum approximation BP decoding algorithm, by replacing the multiplication operation with simple sign, comparison and subtraction operations, the problem of high computational complexity caused by setting the scaling factor is solved.

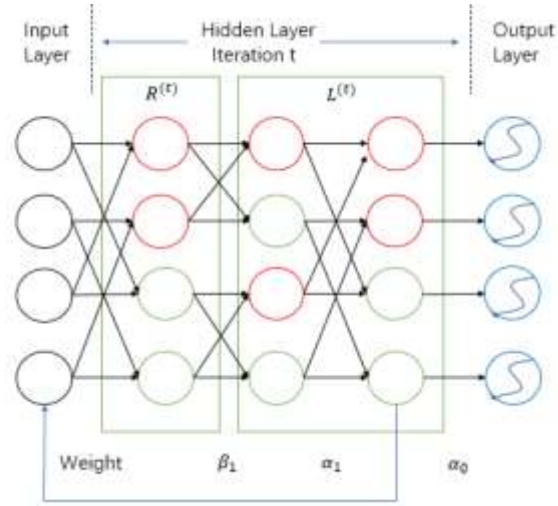


Figure 5. A complete decoding iteration in a recurrent neural network with polar $^{(4,2)}$.

Figure 5 shows a complete iterative decoding process of the $(4,2)$ polar codes in the recurrent neural network, which is expanded according to the polar codes factor graph shown in Figure 2, in the recurrent neural network architecture of Figure 5, $N=4, n=\log_2 N = 2$, so the input layer consists of $N=4$ neurons, and the initial value of the channel log-likelihood ratio is input, the transfer process of the right information and the left information are expanded into $(n-1)=1$ layer and $n=2$ layers, which consists of $N=4$ neurons .Therefore, a complete iteration translates to $(2n-1)=3$ hidden layers, the last hidden layer for right-to-left information transfer computes the output of the leftmost node of the original factor graph. At the end of T iterations, in order to make the codeword estimate obtained by the neural network decoder in the range of $[0,1]$, the output layer of the last layer of the network needs to use the sigmoid function to rescale the output to the range of $[0,1]$:

$$o_j = \sigma\left(L^{(T)}_{1,j}\right) = \left(1 + e^{-L^{(T)}_{1,j}}\right)^{-1} \tag{17}$$

After T iterations are completed, calculate the error between the output o and the transmitted bit u:

$$L(\mathbf{u},\mathbf{o}) = -\frac{1}{N}\sum_{i=1}^{N} u_i \log(o_i) + (1-u_i)\log(1-o_i) \quad (18)$$

Compared with other neural network architectures, this recurrent neural network architecture has obvious optimization. Firstly, the linear rectification function is added in the design of the minimum offset and approximation algorithm, which is equivalent to adding the linear rectification function as the activation function in the calculation process of each neuron in the hidden layer, which plays a role in speeding up the training speed and avoiding the gradient from disappearing. Secondly, after the operation of every $(2n-1)$ hidden layer is completed, the n-bit LLR value is circularly input to realize the reuse of offset parameter set β.

After the CS is established, the bit-flipping BP decoding process assisted by the recurrent neural network can be start. If the RNN-OMS-BP decoder fails to decode with cyclic redundancy check (CRC), as shown in Figure 7, a candidate is selected from the critical set according to the descending index corresponding to the error rate in the critical set, that is, u_i with a larger value $P_e(u_i)$ is flipped first, such sorting can be done offline without introducing the latency caused by sorting. After bit flipping, BP decoding based on RNN is performed, if the result meets CRC, the decoding process is complete, otherwise, other candidate nodes in the critical set are tried until the CRC passes successfully. In this paper, critical set bit flipping is adopted to minimize error propagation, the BP decoding algorithm of polar codes bit flipping based on recurrent neural network is summarized in Algorithm 1, this algorithm is a combination of training and decoding, and the algorithm is as follows:

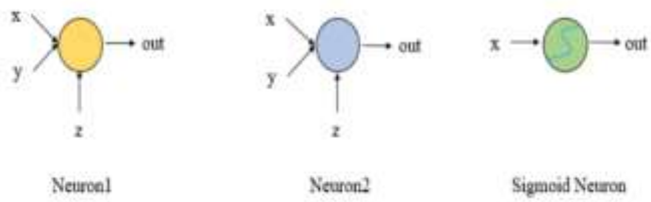


Figure 6. Three types of neurons in network architecture.

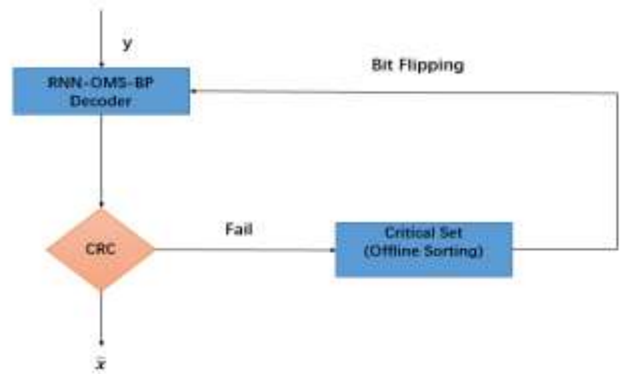


Figure 7. Polar codes bit-flip BP decoder based on RNN.

Algorithm 1 RNN-OMS-BPF algorithm

Input: maximum number of flips $\mathbf{T_{max}}$, critical set $\mathbf{CS}$, N, R, SNR, learning rate η, etc.

Output: $\hat{\mathbf{u}}^{\mathbf{N}}$, $\mathbf{o}$, $BLER$, etc.

for i=0: $\mathbf{SNR}$ **do**
  Generate: $\mathbf{x} = \mathbf{llr}, \mathbf{y} = \mathbf{u}$ // generating a codeword set for training and testing, where x is the generated codeword sequence and y is the transmission bit sequence
end for
for itr=1: epoch (100) **do**
  gendata $(\mathbf{x}, \mathbf{y})$ //get training data
  tf.train.AdamOptimizer($\eta = 0.001$).minimize(loss)
  for i=0: $\mathbf{SNR}$ **do**
    for j=0: $\mathbf{2^{SNR}-1}$ **do**
      L, R L,R ←Initialie the RNN-OMS-BP decoder using (4) and (5)
    calculate the $\mathbf{L}_{\mathbf{i,j}}^{(\mathbf{t})}$ and $\mathbf{R}_{\mathbf{i,j}}^{(\mathbf{t})}$ using (16)
      $\hat{\mathbf{u}}^{\mathbf{N}}$ ←RNN-OMS-BP decoder $(\mathbf{L},\mathbf{R})$
      $\mathbf{t} \leftarrow \mathbf{1}$ // the number of bit flip decoding attempts
      while $\hat{\mathbf{u}}^{\mathbf{N}}$ does not pass CRC && $\mathbf{t} \le \mathbf{T_{max}}$ **do**
        $\mathbf{i} \leftarrow \mathbf{CS(t)}$ //the index in CS
        $\mathbf{R}(\mathbf{A^c},\mathbf{1}) \leftarrow +\infty$ // modify R using (8)
        $\mathbf{R}_{\mathbf{i,j}}^{(1)} = \infty \times \left(\mathbf{2}\hat{\mathbf{u}}_{\mathbf{j}}^{\mathbf{N}} - \mathbf{1}\right), \mathbf{j} \in \mathbf{CS}$
        $\hat{\mathbf{u}}^{\mathbf{N}} \leftarrow 1$ RNN-OMS-BP decoder $(\mathbf{L},\mathbf{R})$
        if $\hat{\mathbf{u}}^{\mathbf{N}}$ pass CRC **then**

```
            return û^N
        else
            if t = T_max then
                The bit-flip BP decoding fails
            end if
        end if
        t ← t + 1
        end while
    end for
    o, Loss = sess.run(x, y, optimizer, loss)  //run the session output prediction bit sequence and loss function value
  end for
end for
```

V. SIMULATION RESULTS AND ANALYSIS

The RNN-OMS-BPF decoder model used in this paper is trained on the deep learning framework Tensorflow 1.8.0 by the adam gradient descent optimization algorithm with a learning rate of 0.001. In order to evaluate the decoding performance of the bit-flip BP decoding algorithm based on the recurrent neural network, a simulation experiment was carried out by the Monte Carlo method, and the all-zero codeword was selected as the training data, the signal-to-noise ranges from 0dB to 3dB, and 3600 codewords are selected as the mini-batch size. The test data is random binary information obtained after polarization coding, binary phase shift keying modulation and transmission on a Gaussian channel, and the weights of the neural network are initialized with random values in a standard normal distribution.

In order to evaluate the BLER performance of the proposed RNN-OMS-BPF algorithm, it is compared with other BP decoding algorithms through simulation, respectively, Table 1 lists the experimental parameters used in the simulation.

TABLE I. SIMULATION PARAMETER

Set options	Value
Test platform	Tensorflow1.8.0
Encoding	$(64, 32)$
CRC Generator Polynomial	$x^6 + x^5 + 1$
Training codeword	7.8×10^4
Testing codeword	3.4×10^3
Batch_size	3600
Loss function	Cross entropy
Optimizer	Adam

A. *Performance analysis*

With the same code length and code rate, the BLER performance of different algorithms is compared with that of the RNN-OMS-BPF decoding algorithm proposed in this paper, the number of iterations of the RNN-OMS-BPF decoding algorithm is 5, and the simulation results are shown in Figure 8.

When the simulation code length is 64 and the code rate is 0.5, the BLER performance of traditional BP, RNN-BP, CS-BF, CNN-Tree-MBF, CA-SCL and RNN-OMS-BPF decoding algorithm proposed in this paper are compared and analyzed. First of all, we can observe that for $(64, 32)$ polar codes, the $BLER$ performance of traditional BP decoding algorithm and RNN-BP decoding algorithm is poor, the RNN-OMS-BPF decoding algorithm proposed in this paper has achieved good decoding performance and the BLER performance is better than that of CA-SCL decoding algorithm with $L=4$. When the number of iterations of BP decoder reaches the maximum and the current decoding result fails the CRC check, the bit flip decoding starts. The bit flip CS-BF algorithm realizes bit flip in BP decoding by setting infinite prior LLR for unreliable information bits in CS set. The simulation results show that, compared with the traditional BP decoding algorithm of polar codes, the bit-flip CS-BF decoding algorithm with bit-flip $\omega=3$ has a BLER performance gain of about 2dB when the BLER is 1×10^{-1}.

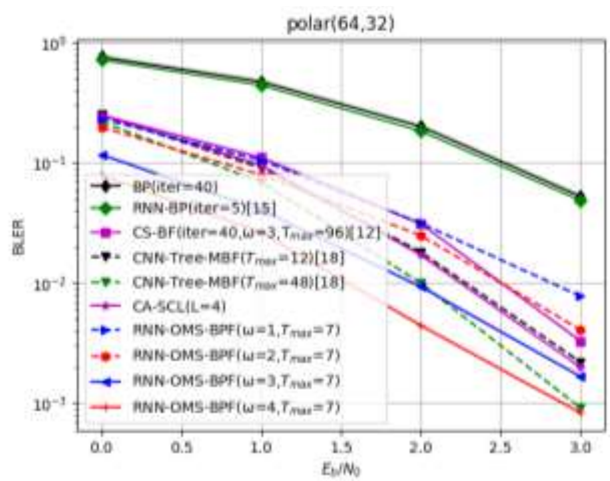


Figure 8. BLER performance of different algorithms.

In order to speed up the convergence rate of the decoding algorithm, reduce the number of iterations required to achieve the convergence effect, and improve the computational resource consumption and extra memory overhead of the traditional BP decoding algorithm, this paper designs a recurrent neural network architecture on the basis of bit flipping, and combines the critical set to perform bit flipping, and selects a candidate bit from the critical set to flip according to the index in descending order of the corresponding error rate in the critical set. Based on the BP algorithm, the BP decoding algorithm is improved by adopting the Offset Min-Sum OMS. According to the similarity between the polar codes BP decoding factor graph and the neural network structure, the polar codes BP decoding factor graph is expanded on the basis of RNN architecture to form a recurrent neural network decoder, and all zero codewords are generated. The recurrent neural network decoder is trained by using the back propagation and Adam gradient descent optimization algorithm in deep learning technology. Compared with CS-BF decoding algorithm, the RNN-OMS-BPF algorithm proposed in this paper has a performance gain of about 0.50dB when the BLER is 3.407×10^{-3}, however, compared with CS-BF decoding algorithm and CNN-Tree-MBF decoding algorithm, RNN-OMS-BPF decoding algorithm has a smaller maximum number of flips, the RNN-OMS-BPF decoding algorithm proposed in this paper can significantly improve the error correction ability with fewer flips.

B. *Computational complexity analysis*

The computational complexity of several decoding algorithms recorded in Table 2, taking $(64,32)$ polar codes as an example, the computational complexity of multiplication of various algorithms in the process of iterative updating of core information is listed when the SNR is 0, the number of iterations of RNN-BP decoding algorithm, RNN-OMS-BPF decoding algorithm are 5 times, and CS-BF decoding algorithm is 40 times, the RNN-OMS-BPF decoding algorithm proposed in this paper replaces all multiplication operations at the cost of increasing some addition operations.

TABLE II. COMPUTATIONAL COMPLEXITY ANALYSIS

Decoding algorithm	Multiplication
RNN-BP	$2IN\log_2 N = 3840$
CS-BF	$2IN\log_2 NT_{avg} = 144080$
CNN-Tree-MBF	$2IN\log_2 NT_{avg} + 9.8M \times T_{CNN}$
CA-SCL	$L\left(N\log_2 N + \frac{3N}{2} + \frac{5K}{2}\right) - \frac{1}{2}K = 4464$
Proposed	0

C. *Memory overhead analysis*

The memory overhead analysis in Table 3 takes the $(64,32)$ polar codes as an example, the memory overhead of various algorithms is listed.

TABLE III. MEMORY OVERHEAD ANALYSIS

Decoding algorithm	Memory overhead
RNN-BP	$N(\log_2 N + 1) = 448$
CS-BF	$N(\log_2 N + 1) + \omega T_{max} = 736$
CNN-Tree-MBF	$2IN(\log_2 N + 1) + T_{max} + K + 0.3M \approx 0.3M$
CA-SCL	$N + 3NL - L = 1592$
Proposed	$N(\log_2 N + 1) + \omega T_{max} = 469$

Table 3 uses the number of parameters to evaluate the memory overhead, where the flip bit ω is 3 and the SNR is 0dB. The RNN-BP decoding algorithm stores L, R messages for computing updated messages, approximately $N(\log_2 N+1)$ messages, while the CS-BF decoding algorithm requires additional memory to store the flipped table, approximately ωT_{max} However, CNN-tree-MBF decoding algorithm needs to store L and R messages of each iteration as input data, which is $2I$ times of

RNN-BP, in addition, additional memory is needed for prediction results, decoding bit values required parameters. The RNN-OMS-BPF decoding algorithm proposed in this paper significantly improves the error correction ability and effectively reduces the memory overhead under the condition of fewer rollover times.

VI. CONCLUSIONS

In this paper, a BP decoding algorithm of polar codes bit flipping based on recurrent neural network is proposed, by selecting bits from the critical set of bit flipping constructed by most error-prone bits, the error correction ability can be significantly improved with less flipping times, and the BLER of BP decoding algorithm can be effectively reduced. Moreover, the multiplication operation is replaced by the Offset Min-Sum approximation algorithm, and the improved recurrent neural network architecture is adopted to realize parameter sharing, which effectively reduces the memory overhead and improves the computational resource consumption of traditional BP decoding algorithm. Next, the stop set will be combined to reduce unnecessary flip bits to reduce decoding delay, so as to obtain better performance gain.

to determine the flip direction and storage of CNN

REFERENCES

[1] Arikan E. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels [J]. IEEE Transactions on Information Theory, 2009, 55(7):3051-3073.

[2] Liu C, Dai W, Guo R. Syndrome Check Aided Fast-SSCANL Decoding Algorithm for Polar Codes [J]. KSII Transactions on Internet & Information Systems, 2024, 18(5). DOI: 10.3837/tiis.2024.05.014

[3] Liu H, Gunawan E, Yaoyue H, et al. BP-Based Sparse Graph List Decoding of Polar Codes [J]. IEEE communications letters: A publication of the IEEE Communications Society, 2023(5):27.

[4] Tal I, Vardy A. List Decoding of Polar Codes[C]// IEEE. IEEE, 2011.

[5] Ren, Y. Shen, Y. Zhang, L. Kristensen, A.T.; Balatsoukas-Stimming, A. Boutillon, E. Burg, A. Zhang, C. High-Throughput and Flexible Belief Propagation List Decoder for Polar Codes [J]. IEEE Trans. Signal Process. 2024, 72, 1158–1174.

[6] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, on deep learning based channel decoding, in The 51st Annual Conference on Information Sciences and Systems (CISS). IEEE, pp. 16, 2017.

[7] Lyu W, Zhang Z, Jiao C, et al. Performance Evaluation of Channel Decoding with Deep Neural Networks[C]// 2018 IEEE International Conference on Communications (ICC). IEEE, 2018.

[8] Teng C F, Chen-Hsi, Ho K S, et al. Low-complexity Recurrent Neural Network-based Polar Decoder with Weight Quantization Mechanism. 2018.

[9] Zhang W, Wu X. Low-Latency SCL Bit-Flipping Decoding of Polar Codes [J]. ArXiv, 2023, abs/2306.02629.DOI:10.48550/arXiv.2306.02629.

[10] Teng C F, Ho K S, CH Wu, et al. Convolutional Neural Network-aided Bit-flipping for Belief Propagation Decoding of Polar Codes. 2019.

[11] Xu W, Tan X, Be'Ery Y, et al. Xu W, Tan X, Be'Ery Y, et al. Deep Learning-Aided Belief Propagation Decoder for Polar Codes[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2020, 10(2): 189-203.

[12] Yinyou M, Dong Y, Xingcheng L, et al. Improved Segmented Belief Propagation List Decoding for Polar Codes with Bit-Flipping [J]. China communications, 2024, 21(3):19-36.

[13] Zhouyu, Yue L I, Shuangshuang Z, et al. Polar codes based OFDM-PLC systems in the presence of impulsive noise [J]. High Technology Letters, 2024, 30(2):188-198.

[14] Chen C H, Teng C F, Wu A Y. Low-Complexity LSTM-Assisted Bit-Flipping Algorithm for Successive Cancellation List Polar Decoder [C]// ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020.

[15] Zhang J, Wang M. Belief Propagation Decoder with Multiple Bit-Flipping Sets and Stopping Criteria for Polar Codes [J]. IEEE Access, 2020, PP (99):1-1.

Research on Dictionary-Based Word Segmentation Algorithms Using Trie Structure

Boxing Zhang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 1824595833@qq.com

Qinlong Kang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 1142926763@qq.com

Xin Jing
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: jingxin@xatu.edu.cn

***Abstract*—This study investigates dictionary-based word segmentation algorithms, which are essential in Natural Language Processing (NLP). Chinese word segmentation poses significant challenges due to the lack of clear word delimiters in the language. This paper explores the advantages and limitations of dictionary-based segmentation algorithms, focusing on how data structures such as Trie and Double-Array Trie (DAT) can enhance segmentation efficiency. An analysis of Trie and DAT structures leads to an optimization achieving constant-time state transitions. This paper evaluates and compares various segmentation algorithms, including full segmentation, forward maximum matching, backward maximum matching, and bidirectional maximum matching. The inherent limitations of dictionary-based segmentation, particularly its dependence on dictionaries and poor disambiguation capability, are also discussed.**

Keywords-Word Segmentation; Trie; Natural Language Processing; Double-Array Trie

I. INTRODUCTION

Word segmentation [1] is a fundamental in Natural Language Processing (NLP). It forms the basis for tasks such as word vector encoding, part-of-speech tagging, syntactic parsing, and text analysis. Unlike English, where words are clearly separated by spaces, Chinese words appear as continuous strings. Consequently, any NLP task involving Chinese must address the issue of segmenting text into individual words.

Popular approaches to Chinese word segmentation [2] include statistical-based methods and dictionary-based techniques. Statistics-based segmentation methods are more expensive and slower. Dictionary-based segmentation is one of the simplest and most frequently used methods. It only requires the construction of a dictionary and a strategy for matching words against it. Dictionary lookup essentially involves string matching, where a target string is compared to entries in the dictionary based on specific rules. This approach can be categorized into Full segmentation algorithms, Forward maximum matching algorithms, Backward maximum matching algorithms, and Bidirectional maximum matching algorithms.

Though dictionary-based segmentation is not complex, and its disambiguation performance is poor, it has the advantage of being fast. The key to leveraging this advantage lies not in the segmentation algorithm itself but in the underlying data structure supporting the dictionary. Dictionary-based segmentation efficiency depends heavily on the underlying data structure used to represent the dictionary. Traditional approaches relying on linear scans or hash-based lookups may struggle to achieve the speed and scalability required for modern NLP applications. In this context, advanced data structures like Trie and Double-Array Trie (DAT) provide significant advantages:

Efficient matching: Tries enable prefix-based matching with time complexity proportional to the length of the query, making them well-suited for word segmentation tasks.

Optimized memory usage: DAT structures further enhance Tries by reducing memory overhead and enabling constant-time state transitions, which is critical for handling large-scale dictionaries.

The study aims to address the problem of low efficiency in traditional Chinese word segmentation by integrating Trie and DAT structures into dictionary-based algorithms. Specifically, the research seeks to:

Analyze the theoretical advantages of Trie and DAT in the context of Chinese word segmentation. Design and implement segmentation algorithms leveraging these data structures. Evaluate the performance of the proposed methods through extensive experiments, focusing on speed, memory usage, and scalability.

Combining theoretical insights with practical experiments comprehensively explores the optimization of Chinese word segmentation using advanced data structures. The findings aim to contribute to the broader field of NLP by offering efficient solutions to a critical preprocessing step in text analysis.

II. DATA STRUCTURE

A. Trie

Trie, also known as a prefix tree, is a tree-like data structure [3] that represents a deterministic finite automaton (DFA), where each node corresponds to a state representing the prefix of a string. Moving from a parent node to a child node signifies a state transition. A search is completed when the terminal state is reached, or no further transitions are possible. The key idea of a Trie is leveraging shared prefixes to save time at the cost of additional space, minimizing redundant string comparisons, thus reducing query time and improving efficiency.

Trie is particularly suited for tasks such as statistics [4], sorting, and storing large amounts of strings. Each edge in the Trie corresponds to a character, and a path from the root node to a leaf node forms a complete string. Trie structure do not directly store strings at nodes. Instead, they treat a word as the path from the root to a specific node, marking that node as the word's endpoint.

For example, in the Trie shown in Fig. 1, each string is represented by a path. Searching for a word involves following the path starting from the root node. If the search reaches a node with a special marker, it indicates that the string exists in the set; otherwise, it means the string is not present.

1) Construction

A Trie works like a dictionary, with its directory structure mimicking real-life dictionary organization.

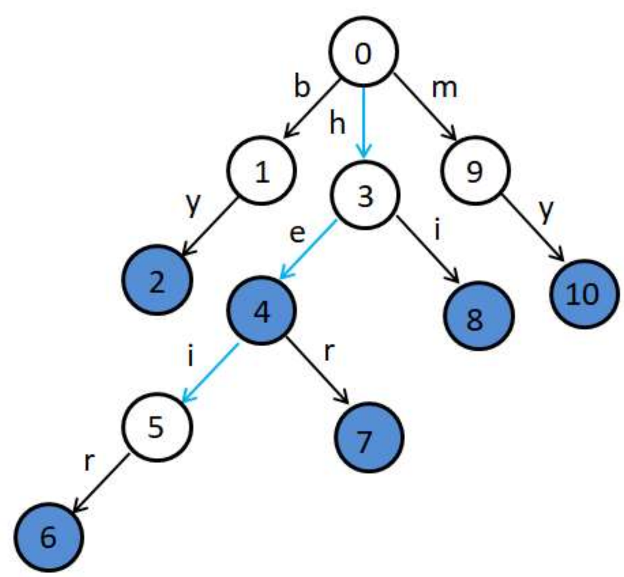


Figure 1. Trie

A string is essentially a path. To query a word, simply follow this path starting from the root node. If it reaches a node with a special marker, it indicates that the string exists in the set; otherwise, it means the string is not present. The paths for string are shown in Table 1.

a) First, a root node is defined, which does not contain any value.

TABLE I. PATH DIAGRAM

String	Path
by	0—1—2
he	0—3—4
heir	0—3—4—5—6
her	0—3—4—7
hi	0—3—8
my	0—9—10

b) Then, using a for loop in a manner similar to Depth-First Search (DFS), each character of the string is checked sequentially to see if it exists in the Trie.

c) If a character does not exist, a child node is created and inserted.

d) If it does exist, the next subtree is retrieved in a DFS manner.

The construction process of the Trie can be broken down as shown in Fig. 2.

The overall steps are as follows:

a) First, define the root node, which does not store any characters.

b) When adding the first string, "by", to the Trie, since "b" does not exist in the Trie, it is added. Since the first character is not present in the Trie, none of the subsequent characters in the string will have been traversed either, so the following characters are added starting from the node indicated by "b".

c) After all characters of the first string are added to the Trie, the current node's isEnd is set to True, indicating that this node marks the end of the added string.

d) This process is repeated until all strings are added.

2) Lookup

Trie lookup involves matching a string by traversing the nodes in the Trie:

a) Start at the root node and search for the first character of the string.

b) If the character is not found, return a negative result, indicating that the string is not in the dictionary.

c) If the first character is found, proceed with a DFS from the matched node. If at any point during this traversal, the character is not found, as described in step b, the matching is considered unsuccessful.

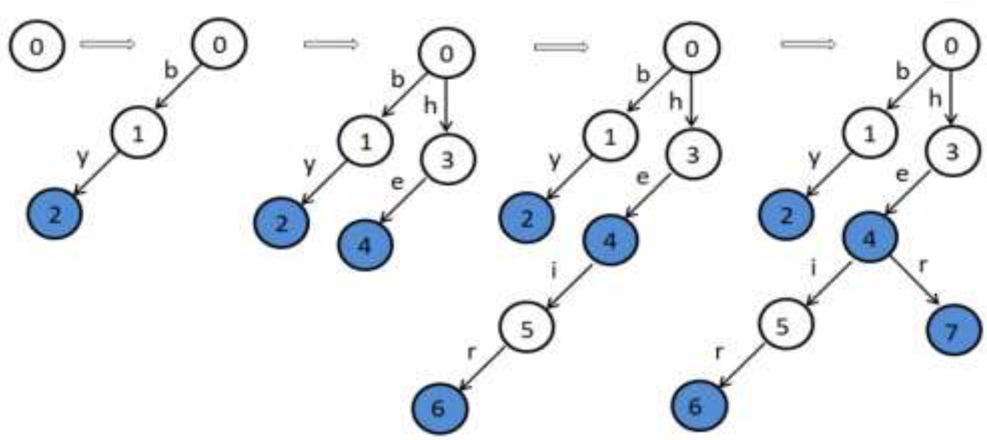


Figure 2. The construction process of the Trie

d) If all characters in the string are matched, check whether the final node is marked as a special terminal node, if isEnd=True, indicating that the full string is present in the dictionary.

In Fig. 1, the blue path shows the matching process for "hei", but since the last node (node 5) is not a terminal node, "hei" is a substring rather than a complete string in the Trie, resulting in a failed match.

3) Performance Evaluation

Constructing a Trie requires scanning all strings, resulting in a time complexity of O(n), where n is the total length of the strings. However, the Trie can be constructed incrementally, allowing for simultaneous querying. The time complexity for querying is O(k), where k is the average length of the strings. Thus, for frequent lookups within a set of strings, the Trie structure is highly efficient.

The key concept of the Trie is to trade space for time. In a typical binary tree, nodes store pointers to left and right children. However, in a Trie, nodes can have many children—up to 26 in the case of the English alphabet. To store these child pointers, an array of pointers (or indices) is used, where each pointer corresponds to a child node. For a set of strings with an average length of l, the worst-case memory usage for storing the pointers in a Trie would be proportional to 26^l, leading to significant memory overhead.

This is particularly problematic in Chinese, where each node needs an array far larger than 26. Even with Java's UTF-16 encoding, a perfect hash requires each node to maintain an array of size 65,536. For a string of length n, the worst-case memory usage can grow exponentially, leading to Out of Memory (OOM) errors. Such a design is impractical for Chinese word segmentation.

What are potential solutions to this issue?

B. Double-Array Trie (DAT)

We apply a binary search strategy to all nodes except the root, enabling the first character match to be achieved with a time complexity of O(1). However, for subsequent nodes using binary search, the state transition complexity is O(log c), where c represents the number of child nodes. When c is large, the transition speed is still slow. In 1989, Jun-Ichi Aoe proposed the Double-Array Trie [5], a data structure with a constant state transition complexity.

1) Algorithm Principles

The DAT is a finite-state automaton [6] that consists of two integer arrays: base and check. Each element in the base array represents a node (or state), while the check array indicates the predecessor state of a node. Initially, all elements in the base and check arrays are set to 0, signifying that a state is unoccupied. If a state represents a complete word, its corresponding base value is set to a negative number (if the state represents a complete word and is not a prefix for any other word, its base value can be set to the negative value of its state position). A state transition is successful when the following conditions are met:

$$p = base[b] + c$$
$$check[p] = base[b]$$

Where *b* is the index of the current state, *c* is the value of the accepted character, and *p* is the index of the transition state. If these conditions are not met, the state transition fails. The state transfer process is shown in Fig. 3.

For example, to check if a transition to the state "西安工" is possible, we compute: "西安工" = *base*[西安] + "工" and verify whether *check*[西安工] == *base*[西安]， This check is performed with one addition and one integer comparison, allowing state transitions in constant time.

2) Construction

The construction of the DAT essentially involves traversing a regular Trie and assigning indices in the double arrays for each node while maintaining and updating the values of the arrays. The most complex part of the construction is determining the base value for each state. The base value of a state must ensure that there is enough space in the array for all its child nodes. The construction process can be outlined as follows:

Take Trie shown in Fig. 4 as an example.

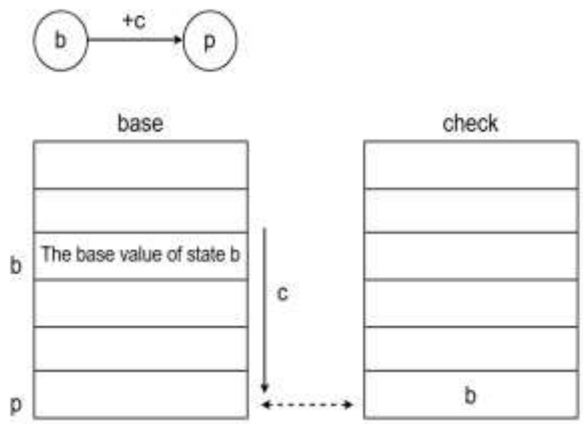


Figure 3. The state transfer process

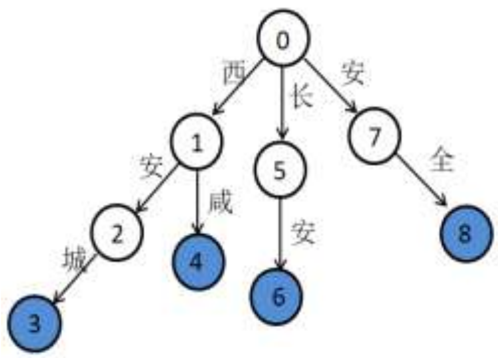


Figure 4. Example of Trie

To simplify, we manually encode the characters in the dictionary as follows, as in Table 2.

a) Initialize the Arrays, as in Table 3

Initialize *base* [0] = 1, with the check array set to all zeros.

b) Assign subscripts to nodes

The root node has child nodes: "西", "长", and "安". We assign positions to them such that *check* [*base* [0] + S_i.code] = 0, meaning the assigned positions are free.

For "西", $S_{西}$.code = 1, *base*[0] = 1, *base*[0]+ $S_{西}$.code = 2. If *check*[2] = 0, assign index 2 to "西", setting *check*[2] = *base*[0] = 1, linking the state represented by index 2 with its parent (the root node). For "长", assign index 6 similarly, setting *check*[6] = *base*[0] = 1. For "安", assign index 3, setting *check*[3] = *base*[0] = 1.

No conflicts occurred during this round of array maintenance, and the results are shown in Table 4.

c) Continued distribution

In the next round, treat the child node "西" as the new parent. First, check whether "西" is a leaf node; since it's not, find its children: "安" and "咸". For "安", $S_{安}$.code = 2, the task is to find a free position for insertion. The transition point is no longer the root node but its parent node "西". Following the rule for calculating the index *p* using the formula *base*[s]+c = p, where is the current node, we have: *base*[2]+$S_{安}$.code = p. There are two unknowns in this equation: the position of *base* [2] and the index *p* for insertion. Let's assume that the insertion index *p* is 4, i.e., p = 4. We can then deduce: *base* [2] = p - 2 = 2. The results at this stage are shown in Table 5.

TABLE II. ENCODE THE CHARACTERS

Character	Code
西	1
安	2
城	3
咸	4
长	5
全	6

TABLE III. INITIALIZE THE ARRAYS

Character											
Serial Number	0	1	2	3	4	5	6	7	8	9	10
base	1										
check	0	0	0	0	0	0	0	0	0	0	0

TABLE IV. START ALLOCATION

Character			西	安			长				
Serial Number	0	1	2	3	4	5	6	7	8	9	10
base	1										
check	0	0	1	1	0	0	1	0	0	0	0

Next, we process the second child node, "咸." Based on the known information, we can calculate *check*[*base*[2] + S $_{咸}$.code] = *check*[6]. Since *check*[6] ≠ 0, this indicates a conflict because the position was already assigned to the character "长" during a previous step. To resolve this conflict, we shift the pre-allocated position by one slot (though it can be shifted by more depending on the algorithm's rules). Now, *check*[7] = 0, so position 7 is allocated to "咸".

To ensure that both *base*[s]+c = p and *check*[p] = *base*[s] are satisfied, we update the arrays. By backtracking, we compute *base*[2] = p - S $_{咸}$.code = 3. Additionally, the state of the previously inserted node "安" must be maintained. Thus, we verify if *check*[*base*[2]+ S $_{安}$.code] = *check*[5] = 0. If this condition is satisfied, "安" is inserted at position 5, and both *base*[5] and *check*[5] are updated accordingly. The array at this point is shown in Table 6.

d) Iteration

And so on, Similarly, it is worth noting that when a node is a leaf node, the corresponding base value is set to -1, as shown in Table 7.

It can be observed that the key challenge in constructing the DAT lies in handling state conflicts. In the process of constructing the DAT, conflicts are inevitable. These conflicts often arise when multiple words share common characters. For example, in the words "西安城", "长安", and "安全", the character "安" is common across all of them. While these words can share a common prefix in the Trie, issues arise when the suffixes contain identical characters, or when suffixes overlap with prefixes. In such cases, a new node must be constructed, which will inevitably cause a conflict. Once a conflict occurs, the base value of the parent node must be adjusted to ensure that all child nodes can find a free position for insertion. This also necessitates the reconstruction of any previously constructed child nodes.

TABLE V. CONTINUE ALLOCATION

Character			西	安	安		长				
Serial Number	0	1	2	3	4	5	6	7	8	9	10
base	1		2								
check	0	0	1	1	0	0	1	0	0	0	

TABLE VI. REALLOCATION PROCESS

Character			西	安		安	长	咸			
Serial Number	0	1	2	3	4	5	6	7	8	9	10
base	1		3								
check	0	0	1	1	0	3	1	3	0	0	

TABLE VII. FINAL RESULT

Character			西	安	安	安	长	咸	城		全
Serial Number	0	1	2	3	4	5	6	7	8	9	10
base	1		3	4	-1	5	2	-1	-1		-1
check	0	0	1	1	2	3	1	3	5	0	4

As a result, the construction time for a DAT can be quite long, and sometimes inserting a single new word may require reconstructing the entire Trie. Additionally, the order in which words are inserted can lead to different conflict scenarios. Typically, when constructing DAT, all the first characters of the words are built first, followed by the child nodes for each word. In this way, if a conflict arises, it can be isolated to a single parent and its immediate child nodes, thus avoiding the need for widespread reconstruction of the Trie.

III. ALGORITHM

A. *Full Segmentation Algorithms*

1) *Algorithm Concept*

The Full Segmentation Algorithms aim to identify all possible words in a segment of text. The logic behind implementing naive full segmentation algorithms is simple: simply traverse the continuous sequences in the text and check whether each sequence exists in the dictionary. The algorithm concept is illustrated in Fig. 5.

The core idea of the Full Segmentation Algorithms involves two for loops that traverse every possible continuous sequence in the text for comparison. The first comparison starts with the first character and then sequentially adds each following character to form a string. If the string exists in the dictionary as a valid word, it is considered a word and added to our list of segmented words. The second loop starts from the second character and combines subsequent characters to form new strings, continuing this process until the last character of the text is reached.

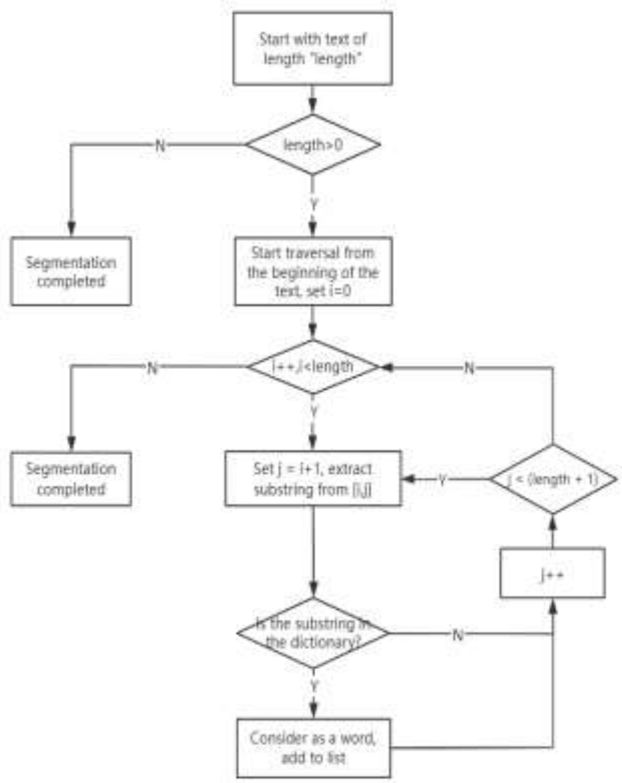


Figure 5. Full Segmentation Algorithm Flow

In the code, an ordered set TreeMap [7] is used, with a complexity of O(log n). The test results are shown in Fig. 6.

As shown, the Full Segmentation Algorithm outputs all individual characters and words from the text. However, this is not the desired outcome for Chinese word segmentation. In practice, what we need is a meaningful sequence of words. For example, we expect "西安工业大学" to be segmented as a single word, rather than fragmented as "西安+工业+大学". This problem occurs for two reasons:

a) The dictionary does not contain "西安工业大学" as a continuous word.

b) The Full Segmentation Algorithm does not account for the fact that longer words typically express meanings that are closer to the actual context.

Thus, we can solve these issues by expanding the dictionary and incorporating the Longest Matching Algorithm.

2) Modifying the Dictionary Word Repository

We can use the mini core dictionary that comes with HanLP. This dictionary is a plain text file that can be opened directly with a text editor. The format after opening is shown in Fig. 7.

The dictionary format in HanLP is a space-separated table. The first column contains the words, and the next two columns represent the word type and corresponding word frequency (the frequency is derived from a corpus). During full segmentation, it was observed that the dictionary does not contain "西安工业大学" as a continuous word.

By manually modifying the dictionary, "西安工业大学" is added to the dictionary repository, as shown in Fig. 8.

Figure 6. Test results

Now, let's retest the Full Segmentation results, as shown in Fig. 9. As the results clearly show, "西安工业大学" appears in the segmentation output. This validates our solution to the first problem, which involved modifying the dictionary. However, in real applications, it is not feasible to manually create a dictionary based on the text to be segmented. Instead, segmentation is performed based on an existing dictionary, which inherently limits the ability to recognize new words.

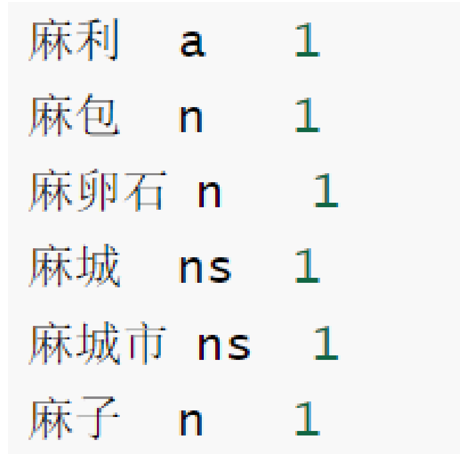


Figure 7. The format

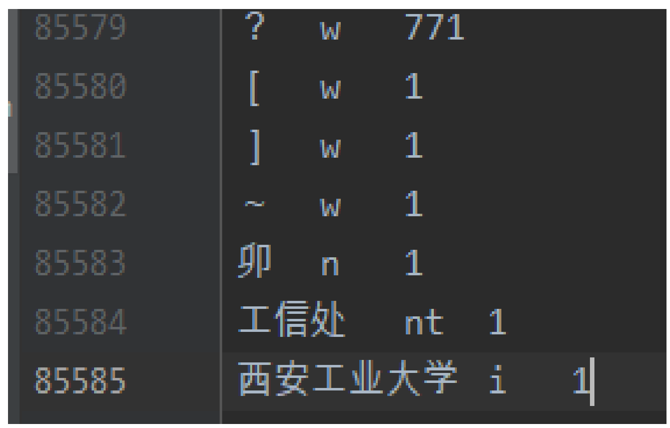


Figure 8. added to the dictionary repository

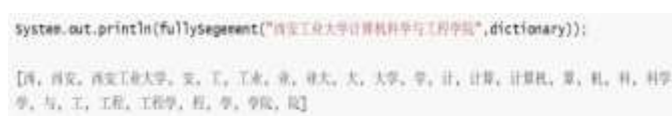


Figure 9. Results After Dictionary Modification

B. Maximum Matching Algorithm

While Full Segmentation Algorithms can capture all the words present in the dictionary, many of the words identified, particularly single characters, are often meaningless. To obtain a more meaningful sequence of words, we introduce the Maximum Matching Algorithm [8], establishing the rule that “longer words take priority”. Specifically, when traversing and matching words starting from a particular index, the longer word is preferred. Depending on the strategy, this leads to three variants: Forward Maximum Matching Algorithm (FMM), Backward Maximum Matching Algorithm (BMM), and Bidirectional Maximum Matching Algorithm (BIMM).

1) Forward Maximum Matching Algorithm

FMM begin by scanning the text from index 0 in a forward direction. In contrast to the logic of the Full Segmentation Algorithm, a new variable *longestword* is introduced to record the longest matched word. The longest word is then added to the word list. The algorithm concept is shown in Fig. 10.

In the FMM, the first *for* loop retrieves the longest string starting from the first character. The second *for* loop then iterates over all possible combinations starting from the first character. If a word exists in the dictionary, the algorithm checks whether the current word is longer than the previous longest match. If it is, the previous match is replaced with the current one. The longest word found in the first round of matching is then added to the list. To ensure the segmentation results, when concatenated, match the original text, the second round of traversal begins from the next character after the longest matched word, and the process repeats until i > text.length(). The implementation result is shown in Fig. 11.

Observation of the output successfully addressed the two main issues encountered with the Full Segmentation Algorithm. However, a new issue arises. In the text “西安工业大学计算机科学与工程学院”, the valid word segments are “工程” and “学院”. Using the Forward Segmentation Algorithm, based on the principle that “longer words take priority”, the segmentation results include “工程学” and the single character “院”, which are clearly incorrect. The error arises because “工程学” takes precedence over “工程” due to its length.

To eliminate this ambiguity, the process can start from the end of the text and traverse forward to find the longest match.

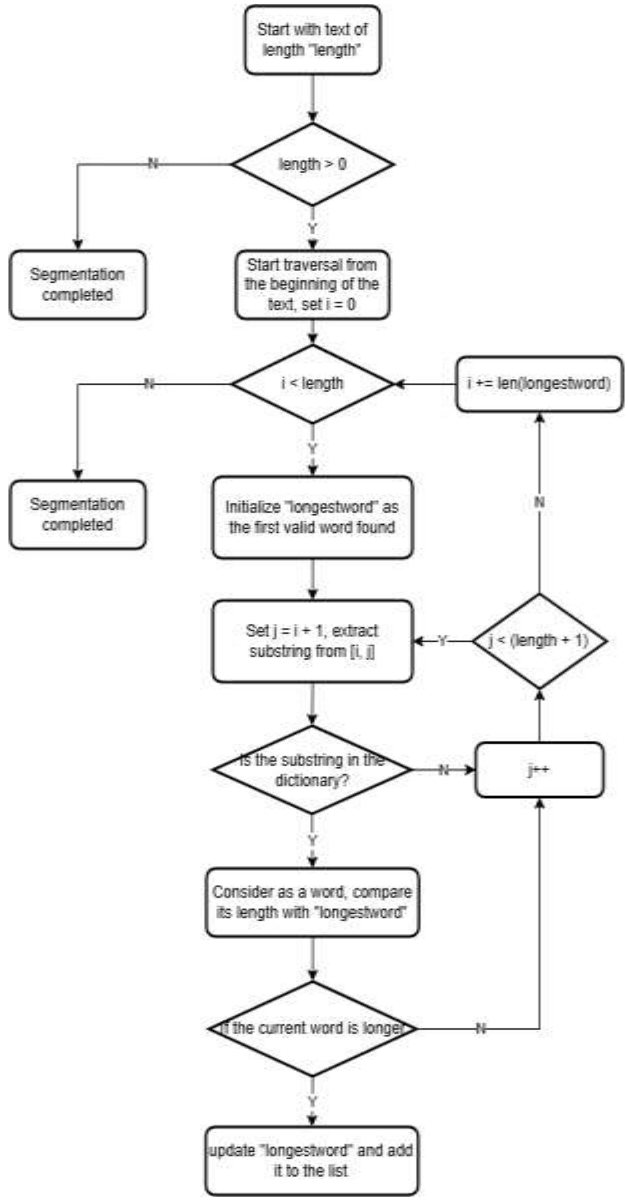


Figure 10. FMM Concept

Figure 11. FMM Segmentation Results

2) Backward Maximum Matching Algorithm

BMM [9] follows the same concept as the FMM, but instead starts from the last character of the text and traverses forward. It’s important to note that does not mean reversing the text; the segmentation results are still ordered according to the sentence’s original sequence. As the algorithm concept has already been described.

Fortunately, the issues caused by the FMM do not appear in the BMM. However, if we use the phrase "醒目的大树" as the segmentation target, the result is problematic:[醒,目的,大树].

This demonstrates that the BMM is not perfect either, and it's difficult to definitively say whether the FMM or BMM performs better.

3) Bidirectional Maximum Matching Algorithm

By applying both the FMM and BMM to segment certain texts, we see that sometimes forward matching performs better, and sometimes backward matching is superior. There are also cases where neither algorithm successfully resolves the ambiguity.

According to Professor Sun Maosong from Tsinghua University, in about 90% of Chinese sentences, the segmentation results of the FMM and BMM are identical. However, in 9% of sentences, the two algorithms produce different results, and one of these results is always correct. Only in 1% of cases do both algorithms fail to produce a correct segmentation result.

This raises the question: is it possible to design a rule-based strategy that selects the correct segmentation result from the two algorithms? In response, the BIMM [10] was proposed, combining both FMM and BMM with the following strategy:

a) Perform both FMM and BMM. If the number of words differs, return the result with fewer words.

b) If the word counts are the same, return the result with fewer single-character words.

c) If the number of single-character words is also the same, prioritize the result from BMM.

The results indicate that while the BIMM successfully selects the best result in certain cases, it chooses incorrect results in others. As a result, its overall accuracy is sometimes even lower than that of the BMM. Therefore, rule-based segmentation algorithms are fragile and cannot guarantee an optimal result, the end result is just robbing Peter to pay Paul.

IV. EXPERIMENTAL RESULT AND ANALYSIS

Dictionary-based Segmentation Algorithms are limited by the dictionary.

A. Naive Implementation

The performance of the four segmentation algorithms was tested under the naive implementation.

1) Definition of Test Methods and Test Cases

a) Speed Test Method

The speed test method is shown in Fig. 12.

```
public static void evaluateSpeed(
    start = System.currentTimeMillis();
            for (int i = 0;i < pressure;i++){
                back_longest_seg(text,dictionary);
            }
            costTime = (System.currentTimeMillis() - start) / (double)1000;
            System.out.println("[illegible]" + text.length() * pressure / 10000 /
                            costTime +"[illegible]");
}
```

Figure 12. Speed test method

First, use the *System.currentTimeMillis* method to record the program's start time and store it in the variable *start*. Then, use a *for* loop to repeatedly perform the segmentation operation *pressure* times. Within the loop, call the *back_longest_seg* method to segment the text *text.*

After the loop completes, record the end time and calculate the total time taken for the entire segmentation process, *costTime*, in seconds. This is calculated by subtracting the start time from the end time and dividing by 1000.

Finally, print the segmentation speed (i.e., the number of characters processed per second). The speed is calculated as *(text.length * pressure) / costTime / 10000*, with the result expressed in units of "ten thousand characters per second".

b) Test Case

Text = "西安工业大学计算机科学与工程学院".

Pressure = 100,000 iterations.

2) Test Results

The test results are shown in Fig. 13.

The results of these tests demonstrated that, under naive conditions, the FMM outperformed the other algorithms in terms of speed. While the

BMM showed competitive performance, the BIMM was slightly slower due to the overhead of combining both forward and backward passes. The Full Segmentation Algorithm was the slowest, given its need to evaluate all possible word combinations, significantly increasing its computational complexity.

Under these circumstances, the time complexity is O (log n).

B. Trie Implementation

We tested the performance of the Full Segmentation Algorithm and the FMM using the Trie structure.

1) *Defining the Test Method and Test Cases*

The testing method and test cases for this structure are similar to those used in the naive implementation.

2) *Test Results*

A Comparison of test results between the two structures is shown in Fig. 14.

C. DAT Implementation

The performance of the Full Segmentation Algorithm and the FMM were tested using the DAT structure.

1) *Defining the Test Method and Test Cases*

The testing method and test cases for this structure are similar to those used in the naive implementation.

2) *Test Results*

The test results of this method are shown in Fig. 15.

The comparison results of the three cases are shown in Fig. 16. The results showed a significant improvement in both algorithms' performance compared to their naive implementations. The DAT structure achieves constant-time state transitions. This reduces overall computational complexity, especially for FMM, where segmentation requires fewer comparisons and faster state transitions.

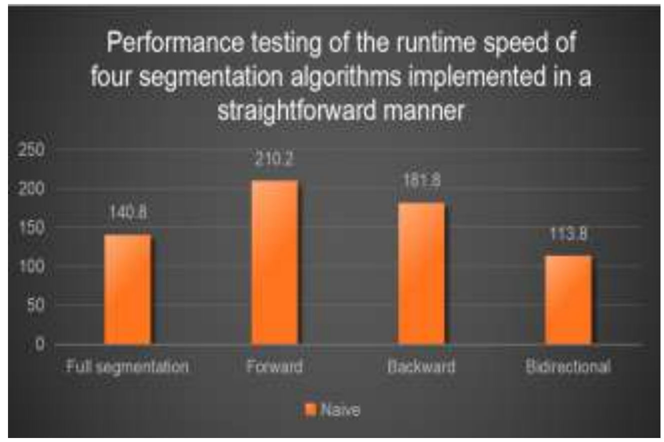


Figure 13. Test results（Naive）

Figure 14. Test results（Trie）

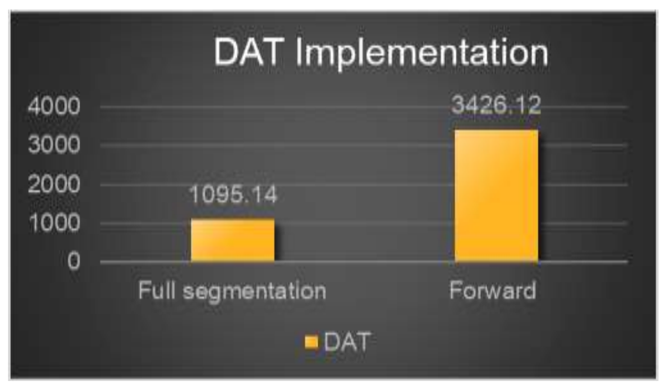


Figure 15. Test results（DAT）

During the state transition process, the time complexity approaches O (1). This is because, in the DAT structure, once the initial state is determined, the transition between states involves only a constant number of operations, such as index calculation and comparison. This efficiency makes the DAT highly suitable for large-scale word segmentation tasks, as it ensures consistent performance regardless of the input size.

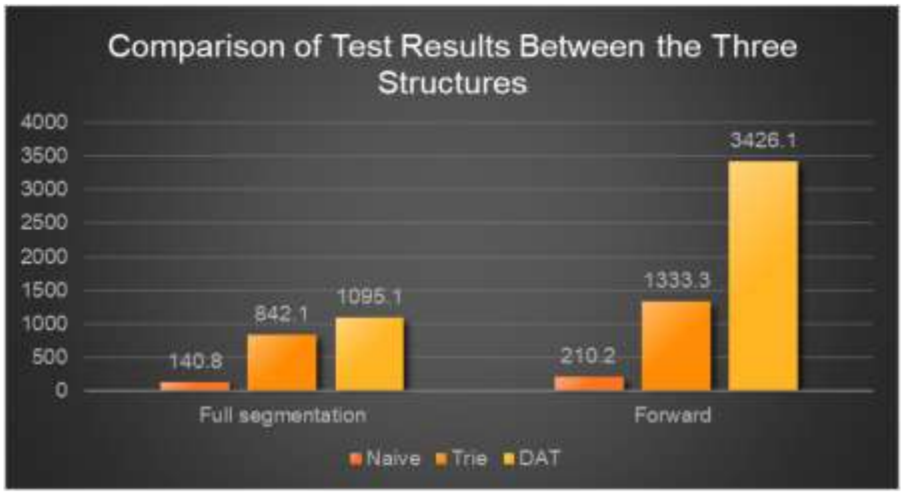


Figure 16. Comparison of test results between the three structures

V. CONCLUSIONS

The segmentation algorithms themselves are not inherently complex. The key to leveraging the natural advantages of fast segmentation lies not in the segmentation process itself, but in the data structure that supports the dictionary. By optimizing the data structure, the efficiency of string matching improves by orders of magnitude.

The DAT achieves constant-time complexity for state transitions, though the algorithm still has limitations:

When performing Full Segmentation on text of length *n*, the complexity can degrade to $O(n^2)$. This is because during full segmentation, the starting point constantly shifts to discover new matches. For example, suppose the dictionary contains all Arabic numerals. Scanning the text “123” results in 6 state transitions: 1, 12, 123, 2, 23, 3. Extending this to the text "123...n," the total number of state transitions becomes $n + (n-1) + \ldots + 1 = n(n+1)/2 = O(n^2)$. An Aho-Corasick Automaton can optimize the DAT by performing only a single scan to find all matches.

On the other hand, it is noted that dictionary-based segmentation heavily relies on the dictionary itself, leading to poor disambiguation and limited ability to recognize new words. Today, in the field of NLP, deep learning-powered statistical models are more prevalent in segmentation algorithms.

REFERENCES

[1] Pak I, Teh P L. Text segmentation techniques: a critical review [J]. Innovative Computing, Optimization and Its Applications: Modelling and Simulations, 2018: 167-181.

[2] Liu C, Zhang Q, Feng J, et al. A Chinese word segmentation method based on dictionary and HMM [C]//Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering. 2022: 644-649.

[3] Sugahara R, Nakashima Y, Inenaga S, et al. Efficiently computing runs on a trie [J]. Theoretical Computer Science, 2021, 887: 143-151.

[4] Yeasin Emon R, Chanda Tista S. An Efficient Word Lookup System by using Improved Trie Algorithm [J]. arXiv e-prints, 2019: arXiv: 1911.01763.

[5] Bannai H, Goto K, Kanda S, et al. NP-Completeness for the Space-Optimality of Double-Array Tries [J]. arXiv preprint arXiv:2403.04951, 2024.

[6] Piedeleu R, Zanasi F. A String Diagrammatic Axiomatisation of Finite-State Automata [C]//FoSSaCS. 2021: 469-489.

[7] Scheibel W, Limberger D, Döllner J. Survey of treemap layout algorithms [C]//Proceedings of the 13th international symposium on visual information communication and interaction. 2020: 1-9.

[8] Pei J. A dictionary-based maximum match algorithm via statistical information for Chinese word segmentation [J]. International Journal of Electronics and Information Engineering, 2020, 12(1): 24-33.

[9] Li R. English Translation Intelligent Recognition Model Based on Reverse Maximum Matching Segmentation Algorithm [C]//International Conference on Innovative Computing. Singapore: Springer Nature Singapore, 2023: 342-349.

[10] Yan X, Xiong X, Cheng X, et al. HMM-BiMM: Hidden Markov Model-based word segmentation via improved Bi-directional Maximal Matching algorithm [J]. Computers & Electrical Engineering, 2021, 94: 107354.

Study and Optimization of Server Load Capacity in High Concurrency Scenarios

Hui Wang
School of Computer Science and Engineering
Xi’an Technological University
Xi’an, China
E-mail: wanghuihy@xatu.edu.cn

Le Qiang
School of Computer Science and Engineering
Xi’an Technological University
Xi’an, China
E-mail: qiangle@st.xatu.edu.cn

Jiasheng Wei
School of Computer Science and Engineering
Xi’an Technological University
Xi’an, China
E-mail: weijiacheng@st.xatu.edu.cn

Haoyu Li
Faculty of Transportation Engineering
Kunming University of Science and Technology
Kunming China
E-mail: haoyumli@gmail.com (H.Y.)

Teng Yan
School of Computer Science and Engineering
Xi’an Technological University
Xi’an, China
E-mail: yanteng@st.xatu.edu.cn

***Abstract*—In high-concurrency scenarios, network and disk I/O-intensive operations often compete for shared resources, resulting in a decline in the server's load capacity. To address this challenge, this paper proposes a sophisticated high-concurrency server optimization solution. It utilizes various Reactor models in the Linux system, combined with the powerful Epoll mechanism and thread pool, to conduct research and optimization on the server's load capacity.Firstly, the event-driven and other modules required by the Web server are implemented and integrated. Secondly, the number of Reactors, the number of threads, and the business processing time under the Linux system are designed and controlled, and the design and implementation scheme of the high-concurrency server based on the Reactor model with the Epoll mechanism and thread pool are determined. Finally, the performance differences and the best usage scenarios of Web servers with different Reactor models in high-concurrency environments are analyzed through stress tests. The comparison results show that the QPS (Queries Per Second) indicator of the Web server based on the multi-Reactor multi-thread model is three times higher than that of the single-Reactor single-thread Web server, verifying its overall advantages in high-concurrency and long-term business processing. The research results demonstrate the applicable scenarios of different Reactor models, providing theoretical basis, implementation examples, and data support for choosing the appropriate Reactor model in actual server development, helping developers select the most suitable Reactor model according to specific server requirements to ensure higher efficiency in high-concurrency scenarios.**

Keywords-component; High Concurrency；Load Capacity；Thread Pool；Reactor model；Web server

I. INTRODUCTION

In the context of the Internet era, which is characterized by an exponential growth in the number of websites and server applications, there has been a significant increase in the volume of data accessed via these platforms. This considerable rise in demand for server resources has consequently resulted in a notable escalation of the load pressure experienced by these servers.

In instances where multiple users access the server simultaneously, the resulting burden on the server's resources is likely to lead to a decline in performance, as well as potential server downtime. An efficacious solution to this problem is the construction of web servers that are capable of effectively managing high concurrency scenarios. The fundamental technology underlying web servers, specifically network programming, has a direct impact on the overall performance of the system.

The most commonly utilized web programming models are the Reactor and Proactor models [2], with the Reactor model being further subdivided into three types based on the number of processes or threads employed. It is important to note that different network programming models are suitable for web servers under varying operational scenarios. Therefore, the aim of this thesis is to investigate the load capacity of servers utilizing different Reactor models in high concurrency scenarios, while also analyzing and summarizing the performance differences between these models. Additionally, the thesis seeks to provide theoretical foundations, implementation examples, and data support for selecting appropriate Reactor models for actual server development. The main work of this article includes:

The objective is to implement the web server modules within the Linux system, which encompasses the following components: event driver, network connection, network processing, thread pool, socket, event, and asynchronous driver.

Each module will be integrated, with the Epoll mechanism and thread pool identified as the foundation for the design and implementation of a high concurrency server based on the Reactor model. Subsequently, the web server will be constructed according to the Reactor model through various combinations and parameter adjustments.

The Web bench stress test tool [3] will be employed to conduct stress tests on web servers utilizing different Reactor models within a highly concurrent environment. The performance of the three Reactor models will then be analyzed using MATLAB, thereby providing a reference for the selection of an appropriate web server programming model based on the performance outcomes of these Reactor models in high concurrency situations.

The remaining sections of this paper are organized as follows:

Section II introduces the relevant technical foundations. Section III introduces the specific design and implementation ideas of the scheme. Section IV introduces the experimental design and result analysis. The conclusions and future work are discussed in Section V.

II. GUIDELINES FOR MANUSCRIPT PREPARATION

A. Technical Architecture for High Concurrency Scenarios

The concurrent access of a large number of users places significant pressure on the server's data exchange and processing capabilities. In order to ensure the successful completion of business operations, a variety of highly concurrent processing techniques have emerged, which are tailored to different application scenarios and present a range of technical architectures. In the context of the civil aviation passenger service information system, Li Yongjin et al. proposed a phased event-driven architecture with the objective of enhancing the system's capacity to process highly concurrent requests [4]. In a further development of WeChat, Li et al. employed Redis caching technology to enhance the system's concurrency [5]. Yuntao Xu et al. employed the use of a Nginx reverse proxy, along with techniques such as multi-processing, multi-threading, and multi-core, with the objective of accelerating the parallel search process in a highly concurrent iris recognition system [6]. Li Junfeng et al. conducted a comprehensive and effective analysis of the high concurrency issue in the airline ticket reservation system, identifying potential solutions through load balancing, page optimization, cache design optimization and database optimization [7]. Wang Jiye et al. achieved high concurrency in the processing of large-scale heterogeneous sensory data, including reception, parsing and distribution, through the utilization of bespoke data structures and asynchronous I/O multiplexing [8]. In addition to such application-specific high concurrency solutions, numerous scholars have also conducted research and advocated for mainstream high concurrency processing techniques. For example, Yannan Wang et al. delineated high concurrency optimization solutions from five perspectives: the web application front-end, back-end program code, database, web application middleware configuration and server load [9]. Kewei Li provided an overview of high concurrency processing techniques for network links, reverse proxies, application services, data caching, and databases, respectively, in Internet distributed architectures [10].

B. multi-threading technology

In addition to enhancing the concurrency of the system through the utilization of hardware technology, prominent web servers such as IIS, Apache, and Tomcat respond to a considerable number of concurrent requests through the implementation of a multi-threading mechanism [11, 12].

In the literature, Bojin Sun et al. put forth a solution to the resource occupancy and contention problems through the implementation of algorithms, data structures, optimized interrupts, and optimized process and routine scheduling. The literature [14] describes the factors that play an important role in the performance of web servers and proposes a new thread-based server architecture. Asynchronous programming enables the development of services capable of handling millions of requests without saturating memory and CPU utilization, thereby enhancing the I/O capabilities of server systems. Karsten M. et al. present the design and implementation of a flexible user-level M: N threading runtime, constructed from scratch, which has been developed to achieve these objectives in [16]. The system's principal components are efficient load balancing and user-level I/O blocking. To address the issue of threads being affected by blocking anomalies, namely the loss of parallelism when executing blocking system calls, which leads to low kernel utilization and unnecessarily high response times, Florian Schmaus et al. introduced pseudo-blocking system calls based on modern asynchronous queuing system call techniques (e.g., Linux's io_uring) in the literature [17] in order to circumvent these anomalies. Techniques such as Nginx and Keepalived are frequently employed to address the load challenges encountered by highly concurrent applications. Literature [18] assesses the performance of a server cluster environment based on Nginx and Keepalived, evaluates the efficacy of Nginx-based algorithms such as WRR, IP_HASH and LEAST_CONN, and designs and optimizes the IP_HASH algorithm.

C. Reactor model

The Reactor model represents an event-driven design pattern that is widely employed in web server development with the objective of creating highly concurrent and high performance web applications. The fundamental concept is to monitor and disseminate input/output (I/O) events via an event distributor (Reactor) and relay the events to the designated event processor for processing. The Reactor model attains efficient concurrent processing through non-blocking I/O and event notification mechanisms, rendering it suitable for highly concurrent, low-latency application scenarios.

The Reactor model may encounter performance bottlenecks and scalability challenges when processing a large number of concurrent connections. The performance and scalability of the Reactor model may be enhanced through the optimization of event processing, scheduling algorithms and I/O multiplexing techniques. The advancement of asynchronous programming and co processing technology has led to the introduction of asynchronous I/O, co-processing scheduler, and other related technologies [21], which have further enhanced the concurrent processing capability and resource utilization of the Reactor model. The Reactor model is currently employed not only in the development of standalone web applications, but also in the context of distributed systems. By designing a distributed event-driven architecture and optimizing the message passing mechanism, the Reactor model is employed in the construction of a highly reliable and high-performance distributed system [22].

The literature [19] examines the advantages, limitations, and applicable scenarios of the Reactor model through an analysis of the event-driven programming model, the design of the event loop, and the registration and distribution of event processors. With regard to the Reactor programming model, domestic and foreign research institutions and enterprises have developed numerous frame-works and libraries with the objective of simplifying the development and maintenance of event drivers. For example, Node.js and Netty [20] represent such frameworks.

The Reactor model typically comprises an event distributor, which monitors input/output (I/O) events and distributes them to the relevant event handlers upon occurrence, and an event handler, which handles the components of a specific I/O event. Event handlers are registered with the event distributor and are invoked upon the occurrence of an event.Three principal models for Reactor modelling in web servers have been identified: the single Reactor single-threading model, the single Reactor multi-threading model, and the multi-Reactor multi-threading model. The suit-ability of different Reactor models for use in various web server environments is a key consideration.

The single Reactor single-threading model represents the most fundamental iteration of the Reactor model, characterized by a straightforward structure, making it well-suited to scenarios involving a limited number of I/O events. The single Reactor multi-threading model builds upon the single Reactor single-threading model by incorporating a thread pool to facilitate event handling and enhance concurrent processing capabilities. The multi-Reactor multi-threading model represents a further optimization of the Reactor model, whereby the concurrent processing capability is enhanced by the distribution of the task of event listening and distribution among multiple Reactor instances. The system is typically constituted of a master Reactor, which listens for connection requests and distributes new connections to slave Reactors. Each slave Reactor runs in a separate thread and listens for I/O events on its respective connection. The slave Reactor then disseminates the events to the worker threads within its management thread pool for processing.

The implementation of multi-Reactor multi-threading represents a departure from the single Reactor multi-threading approach. In this new approach, the Reactor component has been decoupled from a single Reactor module, and instead, it is comprised of a master Reactor module and multiple slave Reactor modules. Concurrently, the original discrete Reactor module is tasked with event listening and distribution, but has also been divided into a master Reactor and a slave Reactor module [24]. The master Reactor module listens for events and disseminates them to a slave Reactor, which oversees the events assigned to it and handles them with an event handler. Multi-Reactor multi-threading is illustrated in Figure 1.

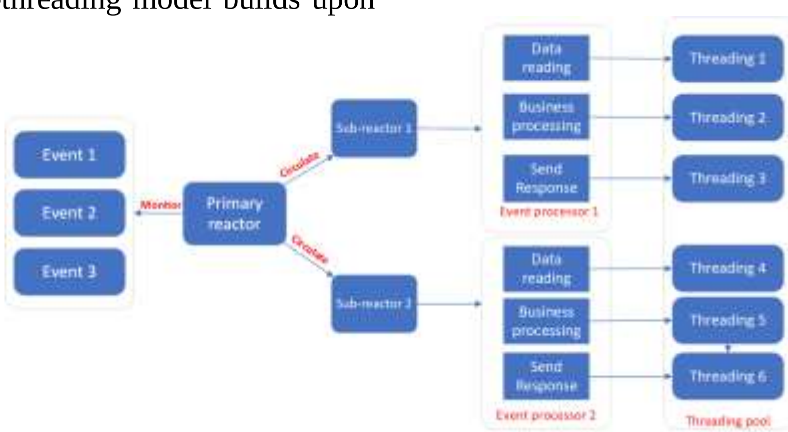


Figure 1. Reactor model based web server

III. DESIGN IDEAS AND SYSTEM CONSTRUCTION

In the Linux system, a web server is designed to control the number of Reactor threads and business processing time, among other variables, in order to simulate different Reactor models. The single Reactor single-threading server serves as the foundation for the subsequent stages of development, beginning with its construction and subsequent evaluation in a highly concurrent environment. This process entails the identification of shortcomings and the implementation of improvements and optimizations to enhance the server model to a single Reactor multi-threaded. The subsequent phase involves rigorous testing and the eventual realization of the multi-Reactor multi-threading model. Finally, the data obtained from the testing phase is subjected to comprehensive analysis and synthesis. Subsequently, the data obtained from the testing of all models will be analyzed and summarized, after which the load capacity of the server in high concurrency scenarios will be studied.

The specific implementation ideas of this paper's scheme are as follows: firstly, the various modules of the web server must be implemented, including the event driver module, network connection module, network processing module, thread pool module, socket module, event module and asynchronous driver module. Secondly, the thesis employs a thread pool to manage multiple threads, thereby enabling the server to transition between a single-threading and a multi-threading model by activating or deactivating the thread pool. Subsequently, the maximum number of concurrently active threads that the thread pool can accommodate is determined by setting the number of threads, thereby enabling an investigation into the impact of varying thread numbers on the performance of a multi-threading web server under high concurrency. Ultimately, the modules are integrated to create three reactor models of web servers through a combination of disparate configurations and parameter modifications.

A. system architecture

Event-driven programming is employed in the construction of high-performance Web servers. These servers are designed to remain continuously attentive to network connection requests, and upon the establishment of a connection, the server will initiate the relevant event processing function to facilitate the processing of the network connection. Once the process of establishing a network connection has been completed, the client socket must be obtained and the corresponding event handler must be triggered in order to read the information from the client. Subsequently, the event handler is triggered in order to respond to the client request.

The Reactor model represents the implementation of event-driven programming concepts within a Web server design pattern. The system is equipped with an event loop, which is responsible for listening to and distributing events. Upon the occurrence of an event, the relevant processor is duly informed and tasked with handling the event.

The Reactor module of the server is implemented by the event driver as the core component. The Reactor module is designed to constantly listen for client requests. Upon detecting a client request, the Reactor is called to handle the connection request event. This enables the establishment of a connection between the server and the client through the Linux kernel, as facilitated by the API for the configuration of client sockets. Consequently, the Reactor module represents the connection between the two entities. Subsequently, the Reactor module will process further events pertaining to the connected client, calling upon the module that has been specifically designed to serve the client in question with the requisite services. Subsequently, the Reactor module will persist in monitoring for new client connection requests and disseminating them to clients who have successfully established a connection. To facilitate the module's service to the client, it will receive the data from the client through the API provided by the Linux kernel, generate the data corresponding to the client after service processing, and transmit the generated data to the client through the API provided by the Linux kernel once more. Figure 2 illustrates the configuration of the Reactor model web server.

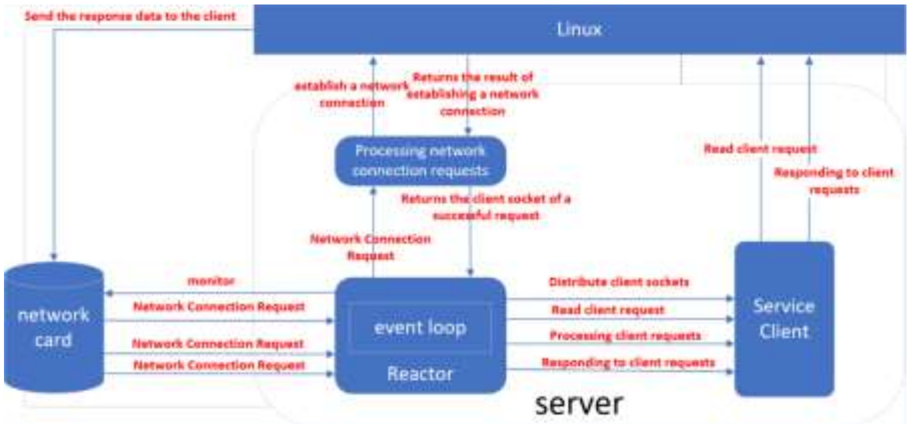


Figure 2. Reactor model based web server

Reactor implementations The development of the Web server is carried out under Linux, and the implementation of the server's Reactor uses Epoll, an I/O multiplexing interface provided by Linux.

Epoll is an I/O event notification mechanism provided by Linux to listen for the occurrence of events registered on Epoll. When Epoll is enabled, the programme blocks until an event occurs in the listening event, and then it returns to that event. This mechanism allows for efficient I/O multiplexing and is often used to build high-performance web servers.

The Reactor module uses Epoll as an event loop for registering, listening, and distributing events, as shown in Figure 3.

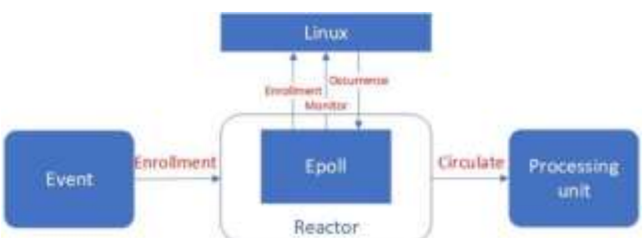


Figure 3. Reactor Modules

Given that Reactor is based on an event-driven model, it can be seen that the key to this model lies in the occurrence and processing of events. Furthermore, given that the web server handles multiple network connection requests simultaneously, it can be seen that this is also a key factor in the event-driven model. The server initially listens to the first client and establishes a connection (listen event), then performs a read event on the first client (read event 1), and finally executes a business process (business process 1) subsequent to the completion of the read event. At this juncture, the second client requests a connection from the server. Consequently, two events are registered on Epoll: the listen event and the write event 1. The order of execution of these two events is indeterminate. The server may execute the listen event first, after which the read event 2 is registered on Epoll. At this juncture, two events are registered on the Epoll (write event 1 and read event 2). The order of execution of these two events is also unknown. This is followed by the occurrence of all subsequent events.

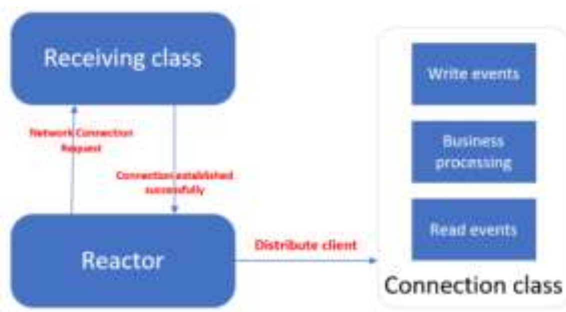


Figure 4. Reactor Modules

The actual test is conducted in a highly concurrent environment, wherein thousands of clients are accessing the web server simultaneously. Consequently, the order of events processed by the server is entirely random. With regard to the type of processor employed in the project, please refer to Figure 4.

B. system implementation

In the implementation of a web server based on disparate Reactor models, it is first necessary to construct the server modules that would be typical of a generic web server. In addition, the specific modules required for each Reactor model must be developed. These include the Reactor module, which implements the Reactor model, and the thread module, which implements the Thread model. In essence, the server implements the web server for different Reactor models by enabling the requisite modules.

With regard to the server module, the fundamental component of the server as a Reactor model is Reactor. Given that Reactor is based on event-driven concepts, it is essential to implement the event-driven module as a preliminary step, and subsequently construct the server module on the foundation of the event-driven module.

The Event-Driven module employs the Event-Driven class as the fundamental component of the event driver, the Epoll class as the implementation of the Event-Driven class for the event driver, and the Event class as the abstract conduit for event interaction with the Event-Driven module to facilitate registration, listening, and updating of events.

1) Event Driver Module

The event-driven module comprises the Event-Driven, Epoll and Event classes, as illustrated in Figure 5.

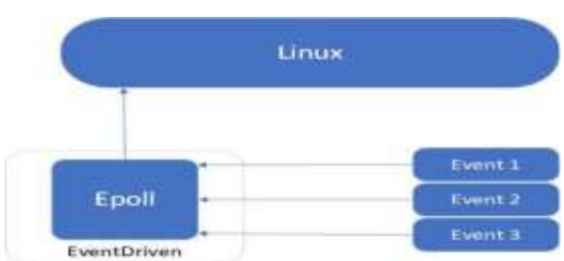


Figure 5. Event Driven Modules

The Epoll class represents an encapsulation of the epoll file descriptor. Epoll is responsible for the registration and updating of events, which it achieves by interacting with the Linux system. Additionally, Epoll listens for the triggering of events. The Event class provides an abstraction of events. The properties of events can be controlled, and interaction with the Epoll class can be facilitated through the Event-Driven class, which is used for the registration and updating of events. The Event-Driven class encapsulates the Epoll class and is employed to initiate the event loop, awaiting the occurrence of events that can be accessed.

a) Event-Driven Class

The Event-Driven class is responsible for event driving, whereby epollfd is employed to encapsulate the I/O multiplexing interface Epoll, provided by Linux. Update Event is used to call the epollfd function of the same name, with the objective of updating the event. The Event-Driven class is illustrated in Figure 6.

Figure 6. Event-Driven Class

b) Epoll Class

The Epoll class encapsulates epoll, which interacts with the Linux system for the purpose of event management. The updateEvent function is employed for the purpose of registering or updating events, while the wait function is used for the purpose of waiting for a registered event to be triggered. The Epoll class is illustrated in Figure 7.

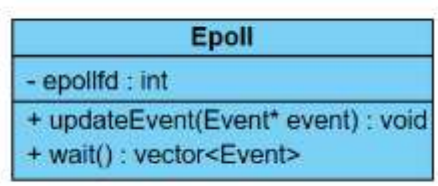


Figure 7. Epoll Class

c) Event Class

The Event class provides an abstraction of an event, utilizing the fd file descriptor to represent a specific event object. It also employs the use of event, inEpoll and callback to represent the type of the current event, the registration status of the event and the event handler function to be executed for the event, respectively. The ed variable is used to associate the Event with Event-Driven, while the latter is used to associate an Event with an Event-Driven. Additionally, ed is employed to pass the Event itself as a parameter to the Event-Driven, thus enabling the execution of the updateEvent function.

The function comprises several parts, including those that set the current event as a listen, read or write event, close the event, check or set whether it is registered, obtain a file descriptor, execute the event handler and set the event handler. The implementation of the code constitutes a direct call to the relevant API provided by the Linux system.

Consequently, the logic related to the execution of the program is not presented, as it is not pertinent to this discussion. The Event class is illustrated in Figure 8.

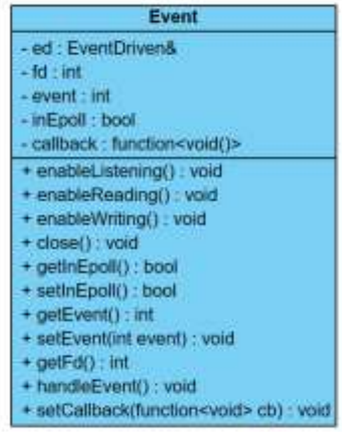


Figure 8. Event Class

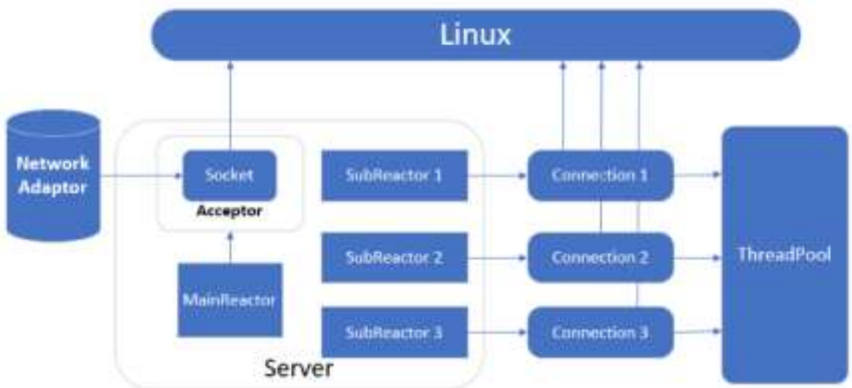


Figure 9. Server Module

2) Server module

The server module serves as the foundation for the implementation of all Reactor-type servers in the thesis. By adding or removing specific modules, it is possible to create different Reactor model web servers. Figure 9 illustrates the server module.

He fundamental component of a server based on the Reactor model is the Reactor module, which is founded upon the principle of event-driven processing. Consequently, the thesis employs the Event-Driven class as the actual Reactor module. The functionality of registering, listening and distributing events inherent to Reactor is achieved through the interaction of the Event-Driven class.

With the Epoll class and the Event class. In the case of the multi-Reactor model, the use of multiple Reactor modules is necessary, with these being divided into Master Reactor and Slave Reactor for different purposes.

The server's function for receiving client requests and establishing connections is designed as the Acceptor class, which interacts with the Linux kernel through the socket class that abstracts sockets to establish a connection with the client.

The Connection class is used to provide specific services to the client, including reading the client's request message, processing the business logic and generating a response message and returning it to the client. In the case of servers operating in a multi-threading mode, the thread pool module is employed.

a) Server Class

The Server class provides the external framework for the entire server model, which serves as a user-facing object and offers methods for initializing the server and initiating the various Reactor models, as illustrated in Figure 10.

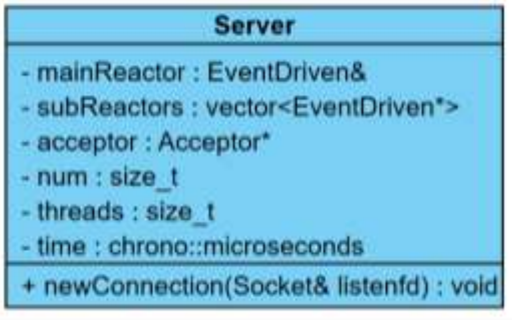


Figure 10. Server Class

The main Reactor module, designated as "main-Reactor," is the primary component of the server. It is responsible for registering, listening to, and distributing all events within the single Reactor model. In contrast, the sub-Reactors module, or "sub-Reactors," operates as a slave Reactor module of the server. In the single Reactor model, it is effectively null. In the multi-Reactor model, the reactor model is empty and receives and processes clients distributed by the main-Reactor. The acceptor is used as an interface to access the Acceptor class. The num parameter is used to specify the number of sub-Reactors, while the thread parameter is used to specify the size of the thread pool. The time parameter is used to specify the length of the business processing time. The new-Connection parameter is used to specify the number of sub-Reactors and the length of the business processing time. The length parameter is used to establish a new client connection.

The server is initiated through the constructor, which first establishes the parameters provided by the user, defining the number of reactors, the number of threads, and the business processing time. Secondly, the acceptor class is initialized, thereby establishing the manner in which the server will handle client connections. Ultimately, the number of reactors serves to distinguish between a single reactor model and a multi-reactor model for the server. In the event that the model is that of a single reactor, the process concludes directly, with the initialization of the slave reactor. The flowchart of the constructor is presented in Figure 11.

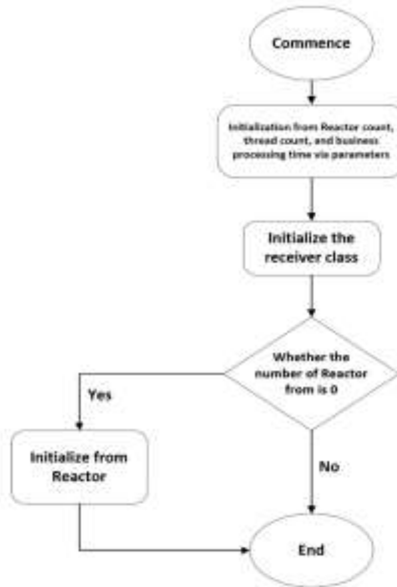


Figure 11. Flowchart of The New-Connection Function

The flowchart of the newConnection function is presented in Figure 12. The responsibility for establishing the connection is allocated to either the slave or the master reactor, depending on whether multiple reactors are employed.

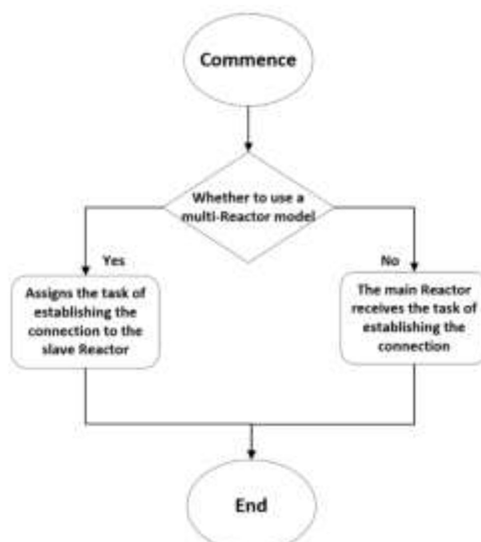


Figure 12. Epoll Class

b) Socket class

The Socket class is analogous to the Event class in that it represents an abstraction of a particular entity, as illustrated in Figure 13. The Socket class provides an abstraction of sockets, where fd represents the file descriptor created by the Linux system to represent the socket. As with the Event class, the functions of the Socket class are all calls to the Linux system APIs, and there is no need to specify the implementation.

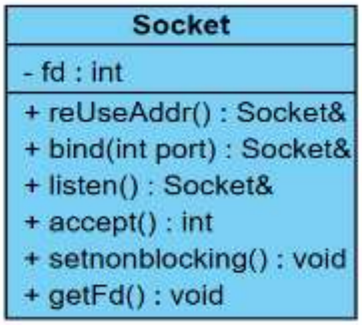


Figure 13. Socket Class

c) Acceptor Class

The Acceptor class is employed to receive new client connection requests and establish a connection, as illustrated in Figure 14. The listen fd socket is utilized as a listening socket to monitor client requests directed towards the server. The event variable represents the current event type, while the callback variable represents the event handler. The setCallback variable represents the event handler. The acceptConnection function is employed to create a connection, which constitutes the primary call listenfd accept function. This is achieved through the utilization of the Linux system API to establish a connection, without the provision of a specific code demonstration.

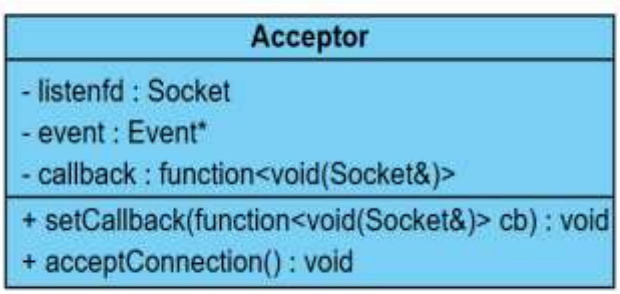


Figure 14. Acceptor Class

d) Connection Class

The Connection class is employed for the purpose of serving clients, as illustrated in Figure 15.

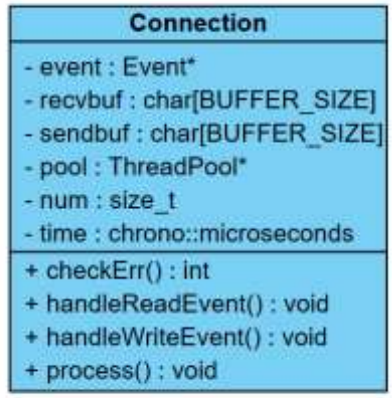


Figure 15. Connection Class

The event represents the current event type. The recvbuf is used to receive the client request message, while the sendbuf is used to send the response message to the client. The pool is the thread pool, while the num is the size of the thread pool. The time is the transaction processing time, and the checkErr is a utility function used to check whether some API calls of the Linux system have returned an error. Finally, the handleReadEvent is used to handle read events, while the handleWriteEvent is used to handle write events. The process is employed for business processing in order to implement the client request. The handleReadEvent is utilized for the handling of read events. The handleWriteEvent is employed for the handling of write events. The process is used for business processing with the objective of fulfilling the client's request. The constructor of Connection will set the event type to read, in addition to initializing the members and binding the event handler to the handleReadEvent. The handleReadEvent is employed for the execution of the read event. With regard to the implementation of the handleReadEvent and handleWriteEvent functions, a number of approaches may be adopted with regard to the reading or writing of data to the buffer. The focus of this paper is on the manner in which the server assigns transactions to the worker threads of the thread pool in a multi-threading model. The flowchart of the handleReadEvent function is shown in Figure 16.

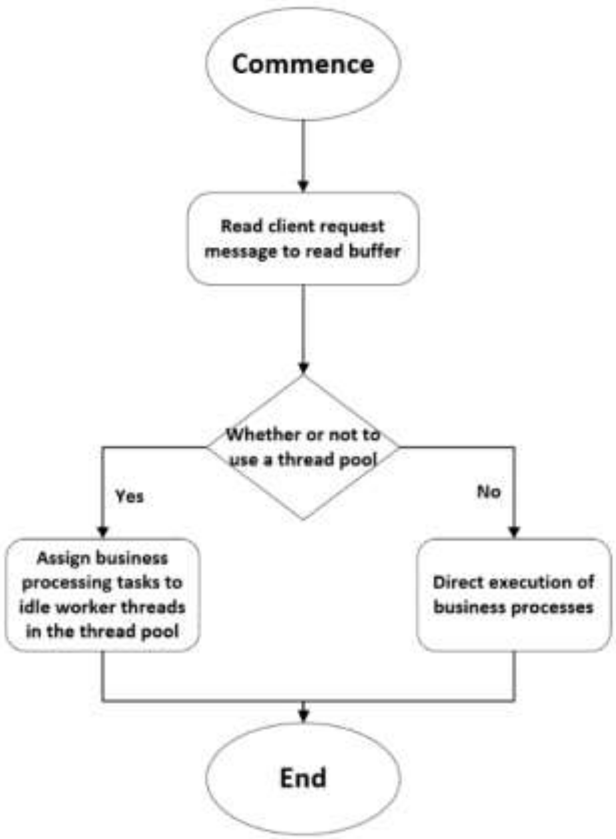


Figure 16. Flowchart of the handle-ReadEvent

Firstly, the message sent by the client is read into a read buffer for parsing and subsequent specific business processing. Secondly, it determines whether the thread pool is to be employed. In the event that the thread pool is to be utilized, the business processing tasks are assigned to the thread pool for completion. Otherwise, the business processing tasks are continued to be executed in the current thread.

e) ThreadPool Class

Figure 17 illustrates the thread pool utilized by the ThreadPool class as a server. The term "works" encompasses all worker threads, while "tasks" represents the functions to be executed. The "mutex" is employed to guarantee synchronization between threads, and the "condition condition variable" is utilized to notify if a new function task has been queued. The "stop" indicator determines whether to halt the pool, and the "enqueue function" is utilized to receive a function to be added to tasks awaiting execution. A function is appended to the list of tasks that are to be executed.

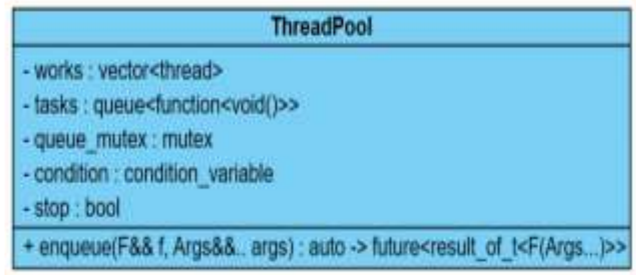


Figure 17. ThreadPool Class

IV. SIMULATION EXPERIMENTS AND ANALYSES

A. Introduction to the experimental environment

The hardware configuration of the test equipment described in the thesis is as follows: the processor is an AMD Ryzen 7 4800H with 8 cores and 16 threads; the memory size is 16 GB at 3200 MHz; and the network card is an Intel® Wi-Fi 6 AX200 160 MHz. The operating system used for the test is Arch Linux.

The test environment is a high concurrency network test environment, utilizing the Webbench stress test tool for high concurrency testing. Webbench is capable of simulating multiple concurrent clients, sending HTTP requests to the server in order to evaluate the server's QPS performance. QPS, Queries Per Second, is a significant index of server performance. It represents the number of network requests that can be processed per second on a web server, and is specifically employed to gauge the server's capacity to withstand high concurrency.

The Webbench test of the web server utilizes a uniform client concurrency of 20,000, requesting the web server's static pages, with an HTTP request time of 5 seconds. The parameters are set in accordance with the specifications outlined in Figure 18.

```
./webbench -c 20000 -t 5 http://127.0.0.1:2000/
```

Figure 18. Using of Webbench

A detailed account of the startup options employed by Webbench is presented in Table 1.

TABLE I. EXPLANATION OF WEBBENCH USAGE OPTIONS

Webbench Usage Parameters	Parameter explanation
./webbench	Starts the Webbench testing tool.
-c	Specify the number of concurrent clients
-t	Specify the duration of HTTP requests
http://127.0.0.1:2000/	accesses the specified web server

The test results for the web server, as displayed in Figure 19, indicate a speed of 81,780 pages per minute and 126,805 bytes per second. The value following 'Speed' represents the number of bytes processed by the web server per second; however, this is not employed as an indicator in the thesis. The value following the designation "Speed" represents the number of bytes per second that are processed by the web server.

Figure 19. Webbench Test Result Chart

A significant variable in the experimental design is the time required for business processing. The business processing time is employed in the modelling of the type and size of network requests. To illustrate, if a client requests a static web page from the server and requests the server to download a file from the server, the input/output (I/O) time spent is different. In most cases, the former is processed more quickly than the latter, which is simulated by the business processing time. A longer business processing time will simulate a longer I/O operation, such as the reading and writing of a large number of files, database queries, and so forth. Conversely, a shorter business processing time will simulate a shorter I/O operation, such as the accessing of some static web pages and small files.

In the web server implementation, the business processing time is set in the business processing events in the connection class, thereby simulating the requisite time for processing the current business by allowing the system to enter a sleep state.

The business processing time is measured in microseconds and is set at server startup; subsequently, it is tested using Webbench. The business processing time is evaluated over a range of 0 to 1000 microseconds, with the server configured to test at 100 microsecond intervals.

The final result for a web server is that the processing time for each service is proportional to the QPS of the current server. This represents the server's performance in handling the current type of service in a highly concurrent environment. Ultimately, the data values (with one business processing time equating to one QPS) are presented in graphical form using Matlab, thus enabling observation of the trend in the optimal performance of the current web server for different business types in a highly concurrent environment.

B. Experimental results and analysis

1) Single Reactor single-threading web server test

The single Reactor single-threading server was subjected to a series of tests using Webbench, with a concurrency of 20000 and a single HTTP request time of 5 seconds. Table 2 illustrates the business processing time for the test variables.

TABLE II. TABLE TYPE STYLES

Variable	Test range
Number of Sub-Reactors	1-50
Number of threads	1-50

The outcomes of the single Reactor single-threading server examination are illustrated in Figure 20. It can be observed that the QPS of the server declines in conjunction with the expansion of business processing time. The most pronounced decline is evident within the 0 – 100 range. It is evident that the performance of the single Reactor single-threading Web server is sub-optimal when processing business that requires a significant amount of time. This is due to the fact that all business processing of the single Reactor single-threading Web server is conducted on a single thread. Consequently, if a business process cannot be completed within the allotted time, it will result in the obstruction of other events, thereby preventing other clients from communicating with the server or causing significant delays.

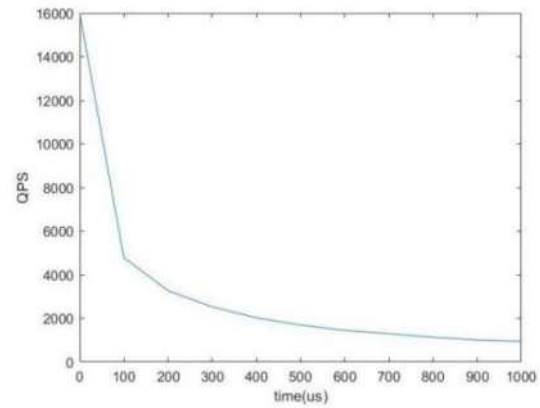


Figure 20. Single Reactor Single Thread Web Server Test Results

2) Single Reactor Multi-threading Web Server Testing

The single Reactor multi-threading server was subjected to testing using Webbench with a concurrency of 20000 and a single HTTP request time of 5 seconds. As illustrated in Table 3, the variables under examination are business processing time and the number of threads. The single Reactor multi-threading mode in which the server operates is dependent on the number of threads. By continually adjusting the number of threads, the optimal QPS value that can be attained by this single reactor multi-threading web server with the optimal number of threads is determined.

TABLE III. SCOPE OF TESTING FOR SINGLE REACTOR MULTI-THREADING WEB SERVER

Variable	Test Range
Number of threads	1-30
Transaction processing time (us)	0-1000

Figure 21 illustrates the single Reactor multi-threading web server QPS, which demonstrates a general upward trend from the bottom right to the top left, reaching a peak, before exhibiting a slight decline and subsequently stabilizing. When the value of TIMES (business processing time) is held constant, there is a notable rise in the QPS of the server as the number of THREADS (threads) increases. This evidence substantiates the assertion that multi-threading effectively addresses the issue of performance degradation caused by single thread blocking.

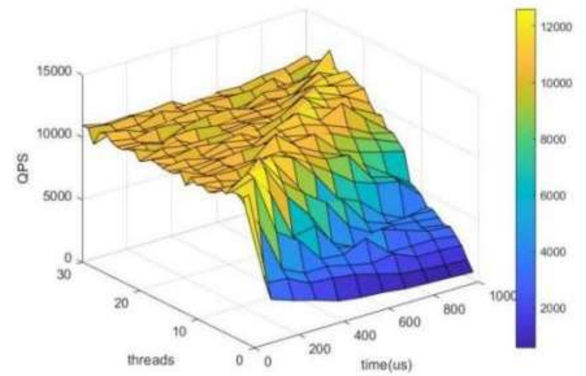


Figure 21. Test Results of Single Reactor Multi thread Web Server

Nevertheless, when a sufficient number of threads are in operation, the server has already reached its optimal performance value. At this juncture, the introduction of additional threads will not yield a positive effect on the optimization of the server. Instead, the frequent switching of threads will result in a reduction in performance.

At this juncture, the optimization component of the thread pool has reached its limit, and the introduction of additional threads does not result in enhanced performance unless the underlying circumstances are altered. The question currently under investigation is what other optimizations are available for single Reactor multi-threading web servers.

3) Multi-Reactor Multi-threading Web Server Testing

The Multi-Reactor Multi thread Web Server should be tested with Webbench using a concurrency of 20000 and a single HTTP request time of 5 seconds. In the context of multi-reactor multi-threading web servers, a comparison is drawn between these and single-reactor multi-threading web servers in terms of business processing time and the number of threads. In addition to these two variables, the reactor number is also taken into account, resulting in a new variable. The number of threads and the number of reactors are combined to create a multitude of potential multi-reactor multi-threading web servers, each with its own distinctive characteristics. To ensure comprehensive evaluation, it is essential to assess the performance of each server under varying service processing times.

TABLE IV. MULTI-REACTOR MULTI-THREADING TEST SCOPE UNDER 200US

Variable	Test range
Operational processing time (us)	0-1000

Table 4 illustrates the results of a performance test conducted on a multi-reactor multi-threading web server with a service processing time of 200 microseconds.

The QPS of the web server was obtained from different combinations of the number of Reactors (subReactors) and the number of threads (threads),

measured when processing 200 microseconds of business data. The results are presented in Figure 22.

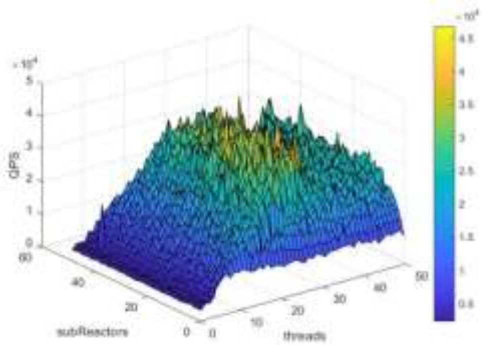
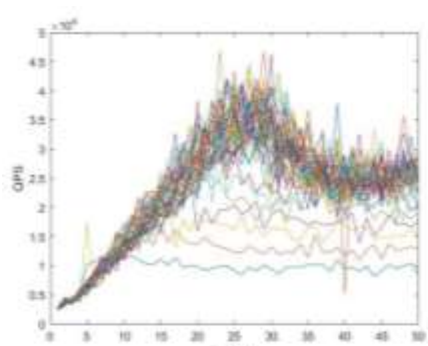


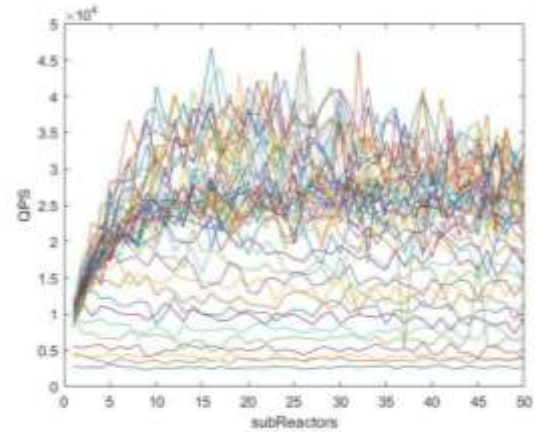
Figure 22. Test Results of Single Reactor Multi thread Web Server

The graph demonstrates a rise in QPS from left to right, reaching a peak and subsequently leveling off. From the outset to the conclusion, the QSP experiences an initial surge and then a decline following a period of sustained peak performance.

As illustrated in Figure 23, the graph is more readily comprehensible when divided vertically into two sections on the X Y axis.: A • m2."



(a) Y-axis section



(b) X-axis section

Figure 23. Cross-section at 200us

As illustrated in Fig. 23(a), the line graph of QPS with the number of threads is fixed with respect to the number of reactors, with each line in the graph corresponding to a specific number of reactors. It is evident that as the number of threads increases, the QPS value rises and reaches a peak; subsequently, as the number of threads continues to increase, the QPS value declines and then stabilizes. It can be observed that after the optimal QPS has been reached, increasing the number of threads in the thread pool does not result in improved server performance. Instead, it has the opposite effect, leading to a decline in server performance. This is due to the fact that an excess of threads causes meaningless thread switching, which results in the waste of server resources and a subsequent reduction in server performance.

As illustrated in Fig. 23(b), a line graph of QPS versus the number of Reactors is presented for a fixed number of threads, with each line corresponding to a number of threads. It can be observed that as the number of slave reactors increases, the QPS of the server exhibits a marked improvement. Further increases in the number of reactors beyond the optimal point have no discernible impact on performance. The continuation of this process resulted in a decline in performance. A notable increase in performance is observed when the number of Reactors opened on the server is compared to the number of Reactors with a smaller number of threads. This increase reaches its peak immediately and subsequent increases in the number of Reactors have a minimal impact on performance. It can be observed that a specific quantity of slave reactors is sufficient to manage high concurrency. The master reactor distributes client services to the slave reactor at a rate of one per second.

The test results for multi-reactor, multi-threading servers under other business processing times are not significantly different. The primary discrepancy lies in the optimal number of slave reactors and the number of threads required to achieve peak server QPS. The trend of server QPS for these tests is analogous to that observed in the aforementioned test, as illustrated in Figure 24.

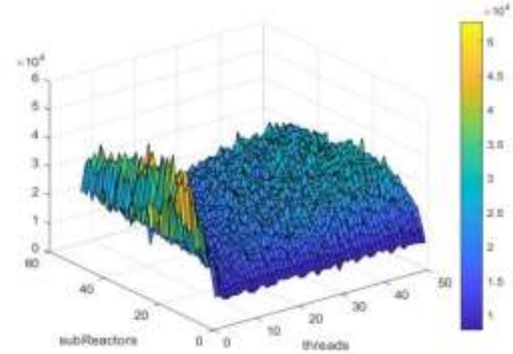


FIGURE (a) Test results at 0us

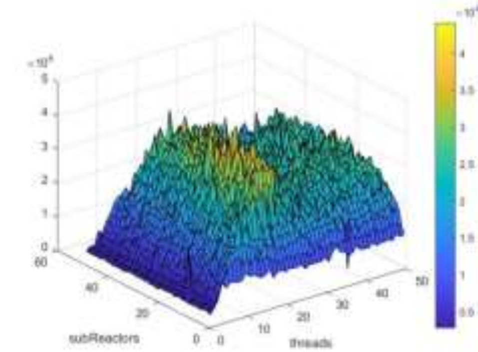


FIGURE (b) Test results at 100us

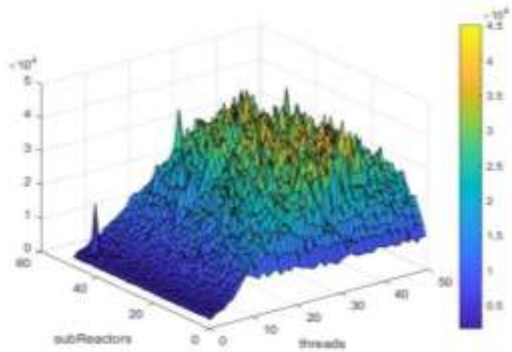


FIGURE (c) Test results at 300us

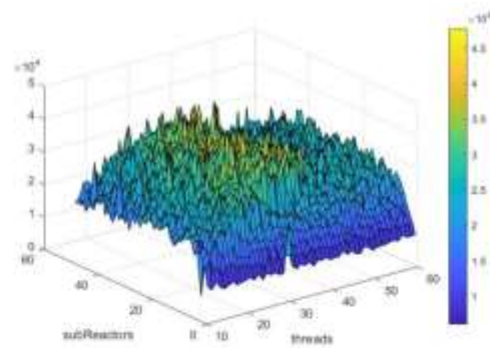


FIGURE (d) Test results at 400us

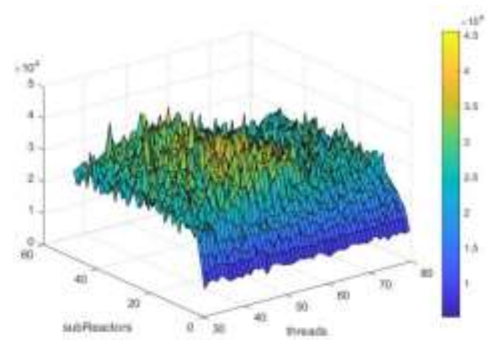


FIGURE (e) Test results at 500us

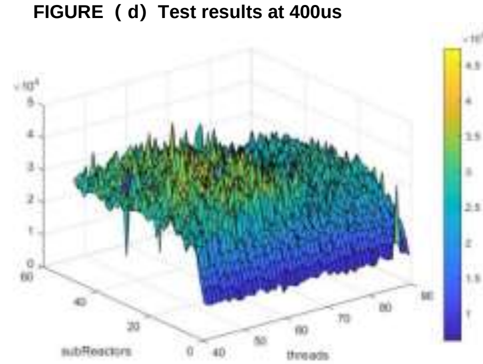


FIGURE (f) Test results at 600us

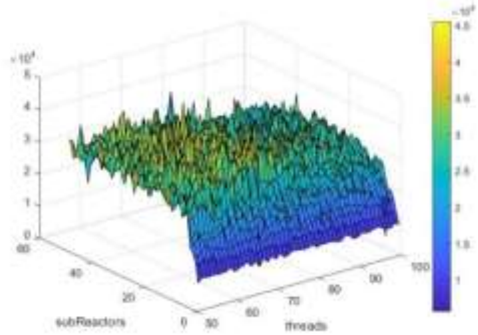


FIGURE （g） Test results at 700us

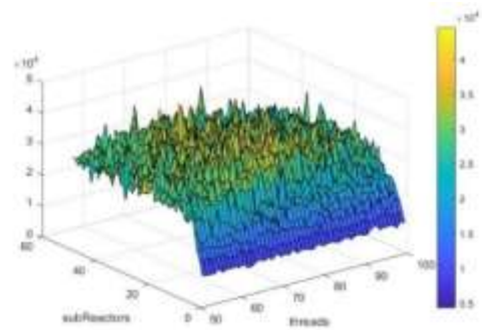


FIGURE （h)Test results at 800us

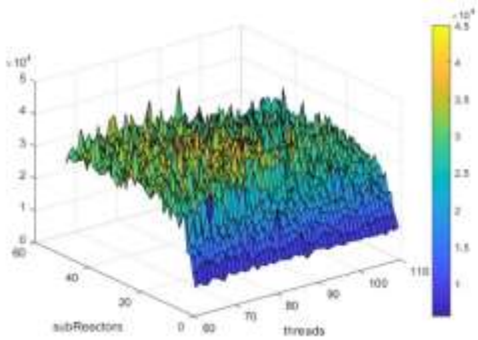


FIGURE （i） Test results at 900us

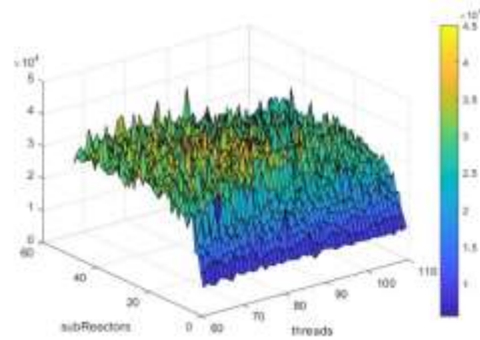


FIGURE （j） Test results at 1000us

Figure 24. Comparison Chart of QPS for Different Time periods

Upon each examination of a server, the source of its performance bottleneck is identified and enhanced, thereby achieving an enhanced performance web server. This process continues until the optimal solution, currently represented by the Multi-Reactor Multi thread Web Server, is reached.

Based on the data shown in the graph, it is evident that the multi - reactor multi - threading web server has significant advantages over the single - reactor single - threading web server. Specifically, the QPS (Queries Per Second) metric of the multi - reactor multi - threading web server is three times higher than that of the single - reactor single - threading web server.

When observing the graph in detail, the single - reactor single - threading web server shows a relatively lower QPS value. Its performance curve rises gradually with the increase of time, but the overall value is not high. This is because, in a single - reactor single - threading architecture, the server can only handle one request at a time. If a new request arrives while the server is processing a previous request, the new request has to wait until the current one is finished. This sequential processing limits the server's throughput and responsiveness, especially when dealing with a large number of concurrent requests.

On the contrary, the multi - reactor multi - threading web server's QPS value shows a remarkable growth. In a multi - reactor multi - threading architecture, multiple reactors are used to monitor different types of events, such as network I/O events. Each reactor can handle multiple threads simultaneously. When a request arrives, it can be quickly assigned to an available thread within the appropriate reactor for processing. This parallel processing mechanism enables the server to handle multiple requests concurrently, greatly improving the processing efficiency.

This substantial improvement in QPS indicates that the multi - reactor multi - threading web server is capable of handling a much larger number of requests simultaneously. In high - concurrency scenarios, where a large number of requests are received in a short period of time, the multi - reactor multi - threading web server can process these requests more efficiently, reducing waiting times and improving overall system performance. The multiple reactors and threads work in parallel, allowing for a greater number of requests to be processed concurrently. Each reactor can handle different sets of requests, and multiple threads within each reactor can execute tasks simultaneously, thereby maximizing the utilization of system resources.

Moreover, the enhanced QPS also implies that the multi - reactor multi - threading web server is better suited for long - term business processing. It can maintain a high level of performance over extended periods, ensuring stable and reliable service delivery. During long - term operation, the single - reactor single - threading web server may encounter bottlenecks due to limited processing capacity, resulting in a decline in performance. For example, if there are a large number of requests queuing up, the single - threaded processing may lead to long - waiting times for some requests, and even cause time - out errors in extreme cases. However, the multi - reactor multi - threading web server can evenly distribute the workload among multiple reactors and threads, avoiding performance degradation caused by long - term operation. This is crucial for applications that require continuous and efficient processing of a large volume of requests, such as e - commerce platforms or large - scale data - intensive services.

In conclusion, the experimental data clearly validates the superiority of the multi - reactor multi - threading web server in terms of high - concurrency handling and long - term business processing capabilities. The significant increase in QPS not only reflects its ability to handle concurrent requests more effectively but also its stability and reliability in long - term operation, making it a preferred choice for web server architectures in high - traffic and data - intensive environments.

The relationship between business processing time and QPS for the final test results of the three Reactor models is presented in a single graph, as illustrated in Figure 25.

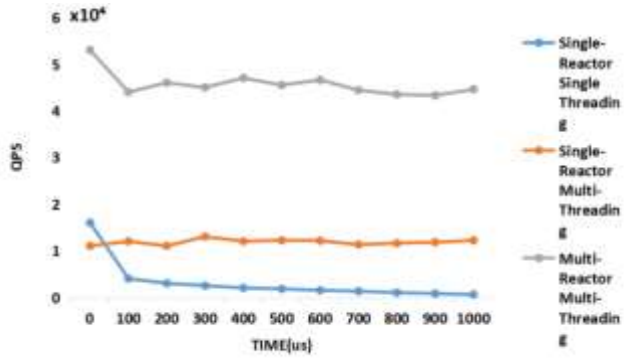


Figure 25. Final Test Results

C. Conclusion of the experiment

The performance of three Reactor models (Single Reactor Single threading, Single Reactor Multi - threading, and Multi - Reactor Multi - threading) in a highly concurrent environment was evaluated, as illustrated in Table 5.

TABLE V. SCOPE OF TESTING FOR SINGLE REACTOR MULTI

Model	Advantages	Disadvantages
Single-Reactor Single Threading	Simple to implement, easy to program and debug; suitable for low concurrency and lightweight business processing scenarios	Poor performance in high concurrency and long time business processing, easy to single thread blocking caused by other requests are delayed processing
Single-Reactor Multi-Threading	The introduction of a thread pooling mechanism serves to enhance the concurrent processing capacity, circumventing the issue of single-thread blocking. This approach is particularly effective in scenarios involving medium concurrency and medium business processing times, offering optimal performance.	The use of multiple threads in a single program can lead to data contention and synchronization problems, necessitating the implementation of locking mechanisms. This, in turn, can result in increased programming complexity and resource overhead. Furthermore, the overhead associated with thread switching may contribute to performance bottlenecks in highly concurrent environments.
Multi-Reactor Multi-Threading	The concurrent processing capability is significantly enhanced by dividing the work among multiple reactors. Each reactor operates independently, reducing competition for resources and improving overall performance. It demonstrates robust performance in high concurrency and long-term business processing.	The complexity of the programming and maintenance processes, coupled with the necessity of dealing with multiple reactor and thread synchronization, gives rise to a considerable challenge in terms of resource management. In order to circumvent performance bottlenecks, it is essential to configure the reactor and thread pool in a reasonable manner.

The Single Reactor Single threading model has a simple structure. However, in a highly concurrent environment, it can only process one request at a time due to having only one thread. This leads to long waiting times as requests queue up, and its performance degrades quickly with increasing concurrency.

The Single Reactor Multi - threading model is an improvement. It uses multiple threads for processing while still relying on a single reactor for event handling. It can handle more concurrent requests compared to the single - threaded model, but the single reactor may become a bottleneck in extremely high - concurrency situations.

The Multi - Reactor Multi - threading model is the most advanced. With multiple reactors and associated threads, it enables a high level of parallelism. Requests can be evenly distributed among the reactors and threads, allowing the system to handle a large number of concurrent requests with short response times. It is highly scalable and suitable for highly concurrent environments like large - scale e - commerce platforms.

These evaluations help in choosing the right architecture for different application scenarios.

V. CONCLUSIONS

In the ever-evolving domain of web server performance, the network programming model, with the Reactor model at its core, has indisputably been a pivotal area of exploration. This comprehensive study has achieved a series of remarkable and significant milestones.

To begin with, it embarked on a detailed investigation into the load capacity of servers that utilize different variants of the Reactor model when confronted with high concurrency scenarios. This entailed meticulously examining how servers with single-reactor single-threading, single-reactor multi-threading, and multi-reactor multi-threading architectures coped under intense traffic loads. Through painstaking research efforts, the study managed to precisely determine the design blueprints and implementation intricacies of high concurrency servers. These were based on the integration of the Reactor model with the highly efficient Epoll mechanism and a well-structured thread pool, all operating within the Linux system environment. The utilization of the Epoll mechanism was particularly crucial as it enabled more efficient event handling and notification, reducing latency and enhancing overall responsiveness.

Subsequently, a series of rigorous stress tests were meticulously carried out on web servers equipped with these diverse Reactor models. The tests were conducted in highly concurrent environments that mimicked real-world, heavy-traffic situations as closely as possible. By subjecting the servers to a barrage of simultaneous requests and analyzing their responses over an extended period, the research team was able to gather a wealth of data. Through painstaking and comprehensive analysis of this data, the final application scenarios for each of the different Reactor models were accurately identified and thoroughly summarized. One of the most significant findings was that, within the constraints and characteristics of the current Linux system, the multi-Reactor multi-threading web server clearly emerged as the preeminent choice for proficiently handling high concurrency requests. It demonstrated superior throughput, shorter response times, and greater scalability compared to its counterparts.

Nevertheless, it is important to acknowledge that, like any research endeavor, this study also has its inherent limitations. The implementation of each model within the scope of this paper was, to some extent, relatively basic and elementary. Crucial elements that are of paramount importance in real-world, practical applications, such as advanced load balancing techniques and sophisticated caching mechanisms, were not given the full consideration they deserved. These factors can have a profound impact on the overall performance and user experience of a web server. For instance, without proper load balancing, servers may experience uneven workload distribution, leading to bottlenecks in some areas while other resources remain underutilized. Similarly, an effective caching mechanism can significantly reduce the need to repeatedly process the same data, thereby enhancing efficiency.

Moreover, the testing regime employed had its own set of constraints. The testing scope was somewhat narrow, primarily focusing on a specific range of concurrency levels and business processing time frames. This meant that it failed to comprehensively encompass all the potential and diverse business scenarios and load conditions that a web server might encounter in the real world. There could be niche applications or extreme traffic spikes that were not adequately represented in the tests, potentially leading to inaccurate generalizations about the performance of the models.

Looking to the future, there is a vast expanse of opportunities and areas ripe for further exploration and research. Future work will be centered around optimizing the design of the multi-reactor model. This will involve delving deep into exploring more efficient thread management strategies. For example, investigating techniques to minimize thread context switching overheads, which can consume significant computational resources and degrade performance. Additionally, the development of more advanced load balancing strategies will be a key focus. This could involve dynamic load balancing algorithms that can adapt to changing traffic patterns and server loads in real-time, ensuring optimal resource utilization.

The model will also be extended to a broader array of real-world applications. In particular, scenarios involving dynamic content processing, such as real-time video streaming or interactive web applications, and large-scale data transmission, like bulk file transfers or database synchronization, will be explored. By applying the model to these diverse settings, its adaptability and effectiveness can be accurately gauged.

Furthermore, it is highly recommended that performance evaluations be carried out across a wide variety of test environments. This would involve incorporating different hardware configurations, ranging from low-end consumer-grade systems to high-performance enterprise servers. Different network conditions, such as high-latency satellite connections or low-bandwidth mobile networks, as well as diverse load modes, including bursty traffic patterns and steady-state loads, should all be considered. By doing so, a more exhaustive and accurate set of performance data can be collected, providing a more comprehensive understanding of the model's capabilities.

Finally, further efforts will be dedicated to developing more efficient synchronization mechanisms. In multi-threading environments, resource contention and excessive thread switching can be major bottlenecks. By devising more intelligent synchronization strategies, such as lock-free data structures or fine-grained locking techniques, the aim is to minimize these overheads and thereby bolster the server's concurrent processing capabilities to new heights. This will ensure that web servers based on the multi-reactor model can meet the ever-increasing demands of modern, high-traffic, and data-intensive applications.

ACKNOWLEDGMENT

The work presented in this paper was supported in part by the Key Industry Chain Technology Research Project of Xi'an Science and Technology Bureau (Grant No. 23ZDCYJSGG0018-2023), the Science and Technology Planning Project (Grant No. GX2412) and the Research project on teaching reform of education in Shaanxi province (Grant No. 23BY082).

REFERENCES

[1] Yuguang Zhang, a layered architecture system for high concurrent processing integrating different scenarios [J]. Communication Technology, 2020, 53(01): 93-100.

[2] Chen R, Mou Y, Li W. A provably secure multi-server authentication scheme based on Chebyshev chaotic map [J]. Journal of Information Security and Applications, 2024, 83: 103788.

[3] Babou N, Boudhar M, Rebaine D. Two-machine job shop problem with a single server and sequence-independent non-anticipatory set-up times [J]. Discrete Optimization, 2024, 53: 100845.

[4] Li YJ, Tian F, Ni ZY. Server architecture design for highly concurrent complex civil aviation services [J]. Computer Applications and Software, 2016, 33(05): 4-7, 39.

[5] Li Jianhua, Xia Flood, Luo Mingquan. Research and implementation of high concurrency WeChat public development based on ThinkPHP and Redis [J]. Computer Application and Software, 2019, 36(02): 108-112.

[6] Yuntao Xu, Wujun Xu, Menglin Zhai. A high concurrency iris recognition system based on B/S

architecture [J]. Computer Engineering, 2019, 45(08): 102-106, 112.

[7] Junfeng Li, Mingxin He. Design and implementation of high concurrency Web airline ticket spike system [J]. Computer Engineering and Design, 2013, 34(03): 778-782.

[8] Wang Jiye, Ding Weilong, Gao Lingchao et al. A sensory data access service supporting high concurrency [J]. Small Microcomputer Systems, 2017, 38(12): 2703-2706.

[9] Yannan Wang, Huarui Wu, Feng Huang. Performance optimisation analysis and research on high concurrency web application system [J]. Computer Engineering and Design, 2014, 35(08): 2976-2981.

[10] Li KW. Practice of high concurrency technology architecture in the Internet [J]. Digital Communication World, 2019(03): 65-66.

[11] Jiexin Zhang, Jianmin Pang, Zheng Zhang, et al. An approach to quantify service quality of mimetic constructed web servers [J]. Computer Science, 2019, 46(11):109-118.

[12] Jiexin Zhang, Jianmin Pang, Zheng Zhang. A method for quantifying web server heterogeneity by mimetic construction [J]. Journal of Software, 2020, 31(2):564-577.

[13] Sun B,Sun M. Concurrency and Operating Systems, Processors, and Programming Languages [J]. Highlights in Science, Engineering and Technology, 2023, 39: 881-887.

[14] Roper M D, Ishihara T, Olsson R A. Critical Performance Factors in Web Server Design: Experience Implementing CoW, a Cooperative Multithreading Web Server [J].

[15] Moslehian A S. An Experimental Integration of io uring and Tokio: An Asynchronous Runtime for Rust [D].

[16] Karsten M, Barghi S. User-level threading: have your cake and eat it too [J]. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 2020, 4(1): 1-30.

[17] Schmaus F, Fischer F, Hönig T, et al. Modern Concurrency Platforms Require Modern System-Call Techniques [J]. 2021.

[18] Ma C, Chi Y. Evaluation test and improvement of load balancing algorithms of Nginx [J]. IEEE Access, 2022, 10: 14311-14324.

[19] Li G, Li J. Optimising low-power task scheduling for multiple users and servers in mobile edge computing by the MUMS framework [J]. Heliyon, 2024, 10(11).

[20] Liu T. Software design of wireless adaptation terminal server in distributed testing system [D]. University of Electronic Science and Technology, 2023. doi: 10.27005/d.cnki.gdzku.2023.000535.

[21] Nie Fanjie. Research and case analysis of high performance server-side framework technology based on Reactor pattern [D]. Zhejiang University of Technology, 2020. doi: 10.27786/d.cnki.gzjlg.2020.000175.

[22] E Qin. Research and improvement of load balancing algorithm based on Nginx high concurrency server [D]. Wuhan University of Technology, 2020. doi: 10.27381/d.cnki.gwlgu.2020.000368.

[23] Ge Y., Li H. Ochre, Li S. Fei. An adaptive dynamic load balancing design and implementation for web server cluster [J]. Computer and Digital Engineering, 2020, 48(12):3002-3007.

[24] Wu Chen. Research and improvement of server cluster load balancing strategy based on Nginx [D]. South China University of Technology, 2020. doi:10.

Research on Classification Method of Film Damage Image Based on Improved ResNet50

Peiqiang Chen
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 1529532803@qq.com

Shuping Xu
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 563937848@qq.com

***Abstract*—In view of the challenges posed by traditional methods in accurately identifying the damage types of film images, this paper proposes an improved approach by leveraging transfer learning based on ResNet50, incorporating the channel attention mechanism from the Convolutional Block Attention Module (CBAM). Additionally, the AlphaDropout model is integrated with the SeLU activation function to enhance stability and performance, and the resultant model is named CBAM-ResNet50. The film dataset employed in this study comprises four types of damage: cracks, dewetting, particles, and scratches. Extensive training and testing of the CBAM-ResNet50 model on this dataset demonstrate significant improvements in classification accuracy. Compared to the original ResNet50 model, the proposed method achieves a remarkable 25.57% improvement in accuracy, reaching 90.58%. This work highlights the potential of combining attention mechanisms and advanced activation strategies to address complex image classification tasks. Furthermore, the approach paves the way for practical applications in quality inspection and automated defect detection in industrial processes.**

Keywords-ResNet50; Convolutional Neural Network; Transfer Learning; CBAM Attention Mechanism

I. INTRODUCTION

Modern optical instruments use a large number of coated optical components. Under the action of strong laser, the damage of optical components is almost inevitable, and once the optical components are damaged, it can disrupt the normal operation of the entire system. Therefore, enhancing the laser resistance of optical films is of great importance [1]. Accurately judging whether the film is damaged or not is to accurately measure the LIDT (Laser Induced Damage Threshold) [2], LIDT is a measure of the laser-induced damage threshold, assessing the laser resistance of optical thin films. Only by accurately measuring LIDT, researchers can explore various approaches to enhance the laser resistance of optical thin films. Thereby reducing the use cost of the optical system.

With the advancement of computer science and technology in our country, Digital image processing technology has also been applied to the field of detecting whether the film is damaged. Compared with the traditional method of using microscope, it has higher accuracy and useability [3-4]. In our country, some scholars began to try to use the method of image classification to study whether the film image is damaged and the type of damage.

Chang Hao et al [5]. Aimed at the problem of film cracking caused by too high or too low temperature in the incubator during film processing. Combining with deep learning, a film rupture image detection technology is proposed, experimental results indicate that the proposed detection technology achieves an accuracy of 82.7%. which can meet the needs of industrial production. Wei Dong [6] proposed two unsupervised surface defect detection algorithms for thin films. The three metrics—precision, recall, and MAP (Mean Average Precision)—are evaluated comprehensively, and the test results are presented. The detection time of single image is 9. 7 ms under GPU acceleration, and the detection speed reaches 103 FPS. Zhang Zhenhua et al. [7] introduced an optical film defect image recognition method utilizing an improved convolutional neural network.

The experimental results demonstrate that the average classification accuracy achieved by this approach is 83.2%, and the training time is 964 s. Liu Peng [8] proposed to develop a set of thin film coating defect detection system to improve the quality of products. After testing, the host computer software functions well and can quickly and accurately process the collected film images. And the detection accuracy meets the required requirements.

In general, image-based methods have been widely used in the detection of thin film damage, However, there are still some problems that need further study. For example, how to improve the accuracy and reliability of detection, how to deal with the interference of complex environment and so on. Therefore, in view of the present situation, this paper presents a film damage image classification method based on an enhanced ResNet50, referred to as CBAM-ResNet50. The experimental results demonstrate a 25% improvement in classification accuracy with the proposed model. 57% compared with the original model, reaching 90. 58%. Moreover, CBAM-ResNet50 has fewer parameters, faster model training speed and shorter training time, which is more suitable for later embedded development. At the same time, it has good generalization ability and can be applied to other types of defect image detection.

II. FILM DAMAGE IMAGE CLASSIFICATION MODEL BASED ON IMPROVED CBAM-RESNET50

A. CBAM-ResNet50 Network Model

He Kaiming and others proposed the concept of residual network (ResNet) in 2015. Since its introduction, ResNet has garnered significant attention in the field of image processing and has delivered outstanding performance on numerous benchmarks. Especially in the ImageNet classification competition, ResNet stands out for its excellent performance. It quickly became one of the most advanced image classification models of its time [9].

Considering the number of parameters and the training effect of each model, this paper chooses ResNet50 model to study. ResNet50 is composed of 50 convolutional layers, including residual blocks, these are directly linked across layers to decrease the number of parameters. Enhance model performance. To boost the classification accuracy of ResNet50, the original ResNet50 architecture is modified. Based on the original model, transfer learning, channel attention mechanism in CBAM attention mechanism, AlphaDropout module and SeLU are used. Activation function to build an improved CBAM-ResNet50 network model. Transfer learning can accelerate the network's training process and help mitigate overfitting issues; CBAM attention mechanism has the ability to screen invalid information, and it can deal with the features extracted from the network model. Only those valuable features are transferred to the next layer, thus optimizing the information transfer efficiency of the whole network; The AlphaDropout module and the SeLU activation function maintain consistency in the mean and standard deviation between the input and output, preserving the normalization property. The network model of CBAM-ResNet50 is shown in Figure1.

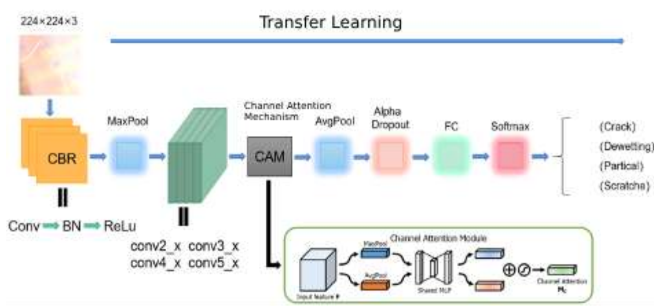


Figure 1. CBAM-ResNet50 Network Model

As shown in the figure, the convolutional and pooling layer structures of the original ResNet50 model remain unchanged. The feature representation of the target image can be enhanced by adding the Channel Attention Mechanism (CAM) after the convolutional layers; An AlphaDropout module and a SeLU activation function are added after the average pooling layer, This ensures that the mean and standard deviation of both the input and output are kept consistent, and maintain the stability of their normalization properties; Because this paper only studies the classification of four types of film damage defects, The Softmax classification layer is thus changed into a Softmax classifier of 4 labels, Corresponding to the four defect types of cracks, dewetting, particles and scratches in the film damage image dataset.

B. Construction of CBAM-ResNet50 network model

1) Transfer learning

When the network is relatively large, that is, the corresponding network parameters are relatively large, and the number of data sets is relatively small. It is not enough to train the whole network, so there will be overfitting, and the training results will be very bad. However, by using the transfer learning method, we can leverage the pre-trained model parameters to train our own smaller datasets. It can also train a better effect [10-11]. Therefore, the method of transfer learning is used in this study. The weights trained on the ImageNet dataset are used to train the film damage dataset. The ImageNet dataset contains over one million images across 1,000 categories, and the number of image categories is sufficient. Although it does not include the film damage image, the judgment of the film damage type is mainly based on its contour and other characteristics. Therefore, transfer learning can be carried out through the ImageNet dataset. The weight file for ResNet50 pre-training using ImageNet can be downloaded from the Tensorflow website.

2) CBAM Attention Mechanism

CBAM (Convolutional Block Attention Module) integrates both channel and spatial attention mechanisms to enhance model performance by emphasizing the importance of different regions in the input image. The channel attention module identifies the most relevant channels in the feature map, while the spatial attention module highlights the spatial locations that contain the most informative content. For the special type of film damage image, its characteristic is that the damage area is often presented in the form of tiling. This means that the distribution of lesions over the entire image space is relatively uniform, so in this case, Increasing the spatial attention mechanism will blur the recognition focus of the model, resulting in a decline in classification accuracy. Based on the above considerations, this study decided to use only the channel attention mechanism to process the film damage images. By increasing the weight of the effective channel and suppressing the weight of the ineffective channel, the model can be designed to focus more on the features that are crucial for the classification task. By more efficiently extracting and utilizing the key information from the image, the model improves its feature representation capabilities, ultimately leading to higher classification accuracy [12-14]. The workflow of the CAM attention mechanism is illustrated in Figure 2.

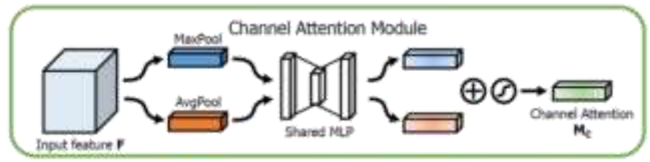


Figure 2. CAM Attention Mechanism Diagram

During the feature extraction process in deep learning, global pooling technology plays a vital role. Firstly, two different pooling strategies, global maximum pooling and global average pooling, are used to aggregate the wide and high dimensional features of the feature vector f, The global maximum pooling operation focuses on the maximum position of each channel in the feature map, it can effectively capture the most prominent features in the image. Global average pooling is a comprehensive perception of each pixel in the feature map to obtain more comprehensive information. Next, the features obtained from global maximum pooling and global average pooling are concatenated. This step aims to fuse the complementary features extracted by both pooling methods, and a richer and more comprehensive feature representation is obtained. Then, the concatenated features are then fed into a multi-layer perceptron (MLP), which consists of two fully connected layers and a ReLU activation function. This MLP is used to further compress and abstract the features through the linear transformation of the fully connected layers and the nonlinear mapping provided by the ReLU activation, High-dimensional feature vectors can be compressed to lower dimensions while retaining information useful for classification tasks. Finally, the output of MLP is normalized by sigmoid function, and the normalized features are merged. A $C \times 1 \times 1$ dimensional channel feature vector is obtained

$M_c(f) \in R^{1\times1\times C}$, $M_c(f) \in R^{1\times1\times C}$ which contains the global information of each wide-high channel feature map $f_c \in R^{1\times H\times W}$. The calculation process is illustrated in formula (1).

$$M_c(f) = \sigma(MLP(AvgPool(f)) + MLP(MaxPool(f)))$$

$$= \sigma\left(W_1(f^c_{max})\right) + W_1\left(W_0(f^c_{max})\right) \quad (1)$$

In (1): W × H — width and height of the feature map; C — number of feature map channels; $f(i, j)$ — feature map; MLP-Multilayer Perceptron.

And finally, multiply that attention map with the weight of the channel corresponding $f_c \in R^{1\times H\times W}$ to the original feature map point by point, Generate the one-dimensional channel feature attention map containing the global feature information required by the spatial attention module f_1, as shown in Equation (2).

$$f_1 = f \otimes M_c(f) \quad (2)$$

3) Alpha Dropout Module + SeLU Activation Function

The original ResNet50 network model has no Dropout module, and the activation function used is ReLU (Rectified Linear Unit).

To address the issue of overfitting, Dropout is applied when the model is large and the number of parameters is extensive. This technique achieves the regularization effect of the model in the training process by temporarily inactivating some neurons. To enhance the model's generalization, however, the distribution of activation values may change after each Dropout application, to solve this problem, researchers proposed AlphaDropout, which is an improvement of Dropout [15]. By using Alpha Dropout, the distribution stability of activation values can be better maintained, and the robustness of the model can be further enhanced. Even in large-scale models and complex tasks, it can effectively deal with the problem of overfitting. At the same time, a new activation function, SeLU (Scaled Exponential Linear Units) [16], is introduced. The combination of AlphaDropout and SeLU ensures that the mean and standard deviation of both the input and output remain consistent, thus maintaining the stability of its normalized properties.

SeLU has a significant advantage over the activation function ReLU in that it does not have a deactivation region, as shown in Equation (3). The SeLU activation function has a saturation region at negative infinity, but this does not adversely affect the expressiveness of the model. In contrast, the SeLU activation function automatically normalizes the sample distribution to 0 mean and unit variance, the stability of the gradient in the training process is ensured, thereby effectively avoiding the problems of gradient explosion and disappearance. This feature of automatic normalization makes the SeLU activation function perform well when dealing with large-scale models and complex tasks.

$$SeLU = \lambda \begin{cases} x & , x > 0 \\ \partial e^x - \partial, & x \le 0 \end{cases} \quad (3)$$

Where ∂ and λ are hyperparameters λ of about 1.05 and ∂ about 1.67 and x are input quantities.

III. EXPERIMENTAL RESULTS AND ANALYSIS

In This experiment, the self-made film damage image data set, ImageNet image data set and Northeastern University steel defect image data set are mainly used. There are four damage types in the film image data set, namely, cracks, dewetting, particles and scratches, which are used for classification and identification. The ImageNet image dataset is used in the transfer learning phase, the steel defect image dataset of Northeastern University is used to verify the generalization ability of CBAM-ResNet50 network model. Accuracy and recognition time are selected as the main evaluation indexes of the network model to facilitate the later embedded development of the model. The model memory footprint and the number of model parameters should also be considered. Combined with the confusion matrix, precision, accuracy, recall, and F_1 score are calculated to comprehensively evaluate the network model's classification ability for thin film damage images.

A. Experimental environment

The experiments were performed on a Windows 10 operating system, featuring an Intel(R) Core(TM) i9-9900 CPU @ 3.60 GHz and 64GB of RAM. The operations were accelerated using an NVIDIA RTX 2080 Ti graphics card. The input images were normalized to a size of 224 × 224 × 3, and the program was written in Python, the model was implemented using the TensorFlow deep learning framework. The initial learning rate was set to 0.0001, the dropout rate was 0.5, the batch size was 32, and the number of epochs was 100.

B. Dataset and Image Preprocessing

1) Film Damage Image Data Set

In this paper, all the original film damage images are numbered, classified and labeled (label 1 is crack, label 2 is dewetting, Label 3 is grain, label 4 is scratch), and finally all the images are imported into the computer in JPG format. The construction of four kinds of film damage image database is completed. All images in this database will be used as the input data set for the CBAM-ResNet50 convolutional neural network. The images in the training set, validation set, and test set are randomly selected by the computer.

2) Steel Defect Image Data Set

The steel defect image dataset from Northeastern University is used to evaluate the generalization ability of the improved CBAM-ResNet50 network model. The images in the steel defect image data set of Northeastern University were collected by several teachers of Northeastern University. The data set includes seven defect types: Rolled-in scale, Patches, Crazing, Pitted, Surface, Inclusion, and Scratches, with 300 images for each defect type. In total, the dataset consists of 1800 grayscale images.

3) Data preprocessing

To remove irrelevant information and preserve the real, useful features in the image, some of the film damage images are corrupted with noise. To enhance the model's generalization ability, Preprocessing the acquired images is crucial. The main pretreatment steps are as follows:

a) Image scaling, ResNet50 requires the size of the input image to be 224 ×224 ×3. Therefore, the size of the acquired image is scaled to 224 * 224 * 3 pixels to adapt to the training of the model;

b) The main function of filtering and denoising is to suppress the noise of the image as much as possible on the premise of retaining the detailed information of the image [17];

c) Edge extraction, which uses edge detection algorithm to extract image features, can effectively remove redundant information in the image. It has certain advantages in improving the classification accuracy [18];

d) data enhancement, to enhance the generalization capability of the network model and prevent overfitting during training, data augmentation of the film damage image dataset is applied. In this paper, the original dataset is divided into three subsets: a training set with 49,760 images, a validation set with 6,220 images, and a test set with 6,220 images, following an 8:1:1 ratio. Only 80% of the training set undergoes data augmentation, while the validation and test sets do not undergo any data augmentation. In this paper, four kinds of image transformation techniques, including rotation, translation, shear and zoom, are used to enhance the images in the training set for seven times. There are 348320 images in the final training set [19];

C. Analysis of factors affecting model performance

1) Comparison of transfer learning effect

In order to avoid the problems of over-fitting and slow convergence, this experiment uses ILSVRC2012 sub-dataset in ImageNet for transfer learning. Figure3 shows the comparison of the relationship curve between the number of iterations of the validation set and the accuracy rate of the CBAM-ResNet50 network model with and without transfer learning. Figure 4 compares the relationship between the number of iterations and the loss value on the training set, with and without transfer learning.

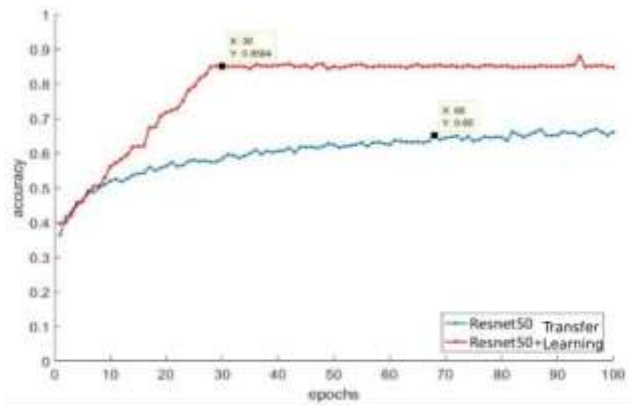


Figure 3. Curve between the number of iterations and the accuracy with or without transfer learning

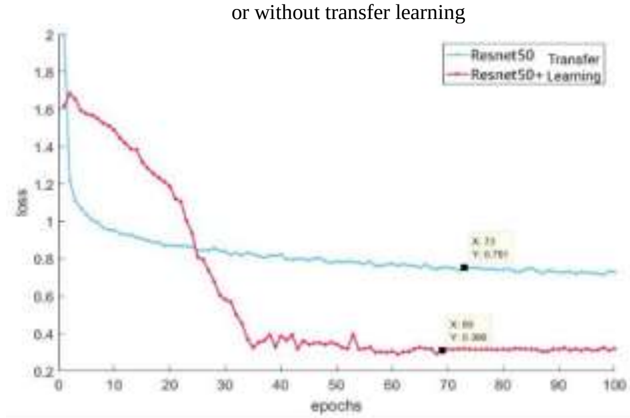


Figure 4. Curve between the number of iterations and the loss value with or without transfer learning

As shown in Figure 3, the ResNet50 + transfer learning model demonstrates higher accuracy and faster convergence compared to the original ResNet50 model on the validation set. Furthermore, the curve becomes more stable after reaching convergence. The ResNet50 + transfer learning model tends to be stable at 30 epochs, while the original ResNet50 model tends to be stable at 68 epochs. The accuracy of ResNet50 + transfer learning model is about 0.85, while the accuracy of ResNet50 model is only about 0.65. This shows that the training accuracy of the network is higher, the training speed is faster, and the network can be stabilized faster after using transfer learning.

As shown in Figure 4, the final loss value of the original ResNet50 network model is relatively high, approximately 0.75. The final loss value of ResNet50 + transfer learning model is 0.4 less than that of ResNet50 +, only 0.3, and the loss curve using transfer learning decreases faster. Achieve stability more quickly.

2) *Effect comparison of CBAM attention mechanism*

In the task of image classification, the internal feature transmission mechanism of the traditional convolutional neural network often lacks clear distinction. As a result, in the process of feature extraction and transmission, effective information is easily submerged in a large number of irrelevant information. This type of feature propagation not only impacts the model's ability to capture crucial information but may also lead to performance degradation in complex scenarios. To overcome this challenge, this paper incorporates the channel attention mechanism from the CBAM module into the traditional ResNet50 architecture. O as to realize the optimization and improvement of the feature transmission process.

The placement of the attention mechanism plays a crucial role in determining the model's recognition accuracy. Feature representations at different levels have different levels of abstraction and semantic information, and therefore, Embedding the attention mechanism in different locations of the network will produce different effects. To enhance the model's performance without altering its original structure, in this paper, we choose to add attention mechanism after the first and last convolution layer of ResNet50.This design can not only ensure that the attention mechanism can fully play its role, but it will not significantly affect the overall structure of the model. To verify the effectiveness of the attention mechanism's placement, four different addition schemes are designed in this paper, as shown in TABLE I.

TABLE I. FOUR CBAM ATTENTION MECHANISM ADDITION SCHEMES

Programmer	***Attention mechanism adding method***
Option 1	Add an attention mechanism after the first convolution layer
Option 2	Two attention mechanisms are added after the first and last convolution layer
Option 3	Add 1 attention mechanism after the last convolutional layer
Option 4	Do not add attention mechanism

During the experiment, the same data set and consistent training parameters are used to ensure the fairness of the comparison results. The results of the experiment are displayed in Figure 5.

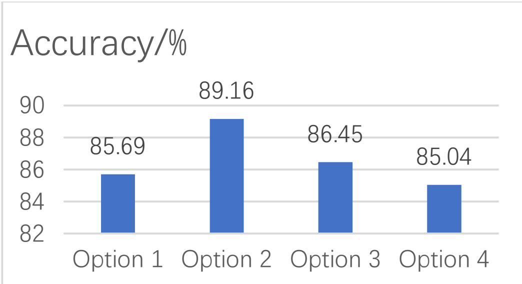


Figure 5. Performance comparison of CBAM-ResNet50 model with different addition modes of attention mechanism

As can be seen from Figure 5, when an attention module is added to the ResNet50 network, its recognition performance is slightly improved. This indicates that the attention mechanism can greatly improve the model's feature extraction capabilities, allowing it to more effectively capture the essential features within the image. When two attention modules are integrated into the network, the recognition accuracy of the film damage images reaches its optimal performance. This outcome strongly validates the effectiveness of the channel attention mechanism in image recognition tasks. The model can mine the potential information in the image more deeply, and further improve its accuracy on the validation set. It is also important to note that while adding more attention modules may result in higher performance gains, but this will also increase the complexity and computational cost of the model. Therefore, in practical applications, it is necessary to balance the performance and complexity of the model according to the specific task and resource constraints. Select the most appropriate number of attention modules and where to add them.

3) Effect comparison of AlphaDropout module + SeLU activation function

In this paper, The Alpha Dropout module and SeLU activation function are innovatively incorporated to enhance the performance of film damage image classification. The combination of these two techniques not only helps to maintain the stability of the distribution of the feature map, it can also effectively prevent the occurrence of over-fitting phenomenon, this approach also accelerates the training process and improves the model's convergence speed.

First of all, the AlphaDropout module is used in the training process. The output of the neurons in the network is set to zero randomly with a certain probability, thereby avoiding the excessive dependence of the model on the output of a specific neuron, the randomness introduced by this approach helps enhance the model's generalization ability. To avoid overfitting, where the model performs well on the training set but shows a significant drop in performance on the test set, the AlphaDropout module is employed. This module effectively improves the model's robustness. It can maintain stable classification performance. Secondly, the SeLU activation function is selected, which has the characteristic of self-normalization. The distribution of the output value can be automatically adjusted during the activation process to be close to the standard normal distribution, This property helps preserve the stability of the feature map distribution, The information loss or distortion caused by the action of the activation function in the layer-by-layer transmission process is prevented, and meanwhile, The SeLU activation function also has a faster convergence rate and can achieve better classification results in a shorter training period.

In order to verify the effect of AlphaDropout combined with SeLU on the classification of film damage images by ResNet50 network, Based On the ResNet50 network using transfer learning and adding CBAM attention mechanism, AlphaDropout + ReLU is tested respectively. AlphaDropout + SeLU and the original ResNet50. The recognition results of the three models under the same conditions are shown in TABLE II. Compared with using ReLU activation function alone, the combination of AlphaDropout and SeLU achieves the highest accuracy of 90.58%. Compared with the combination of Alpha Dropout and ReLU, its accuracy is increased by 0.13%, which is mainly attributed to the activation function of SeLU. It effectively solves the problem of neuron "inactivation", This, in turn, this enhances the expressive capability of the network model. The normalization feature of SeLU, coupled with AlphaDropout, ensures that the output data retains a mean of 0 and a standard deviation of 1. As a result, the model's convergence speed is further accelerated.

TABLE II. PERFORMANCE COMPARISON OF CBAM-RESNET50 MODEL UNDER DIFFERENT ACTIVATION FUNCTIONS

Activation function	*Classification accuracy (%)*
AlphaDropout+SeLU	90.58
AlphaDropout+ReLU	90.45
ReLU	89.16

D. Confusion Matrix of Film Damage Classification and Identification Results on CBAM-ResNet50

The confusion matrix, being an intuitive and effective tool, provides a clear view of the classification model's performance across each category. It can not only reflect the recognition accuracy of the model for each category, but also reveal the confusion of the model between different categories. It provides an important basis for model optimization. In this study, four different types of film damage were classified and identified, and the improved CBAM-ResNet50 model was used to model. To provide a more intuitive illustration of the model's classification performance, this paper presents the corresponding confusion matrix, as illustrated in Figure 6.

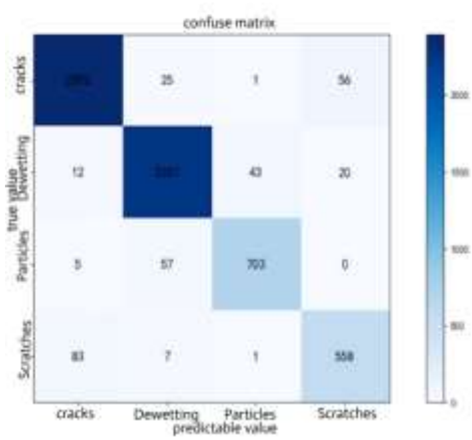


Figure 6. Confusion Matrix of Film Damage Classification Identification Results on CBAM-ResNet50

In this study, the performance indicators of CBAM-ResNet50 model on the task of film damage identification are calculated by using the confusion matrix. As shown in TABLE III.

TABLE III. FILM DAMAGE CLASSIFICATION AND IDENTIFICATION RESULTS OF CBAM-RESNET50 MODEL

Damage category	*Accuracy (%)*	*Accuracy (%)*	*Recall (%)*	F_1 *Fraction*
Crack	95.02	95.99	96.69	96.34
Dewetting		96.21	96.78	96.49
Particles		93.98	91.90	92.93
Scratches		88.01	85.98	86.98

As shown in TABLE III, all metrics, except for the precision, recall, and score of scratches, are above 91%.

As shown in Figure 6, the total number of misrecognitions is relatively low. which indicates that the improved CBAM-ResNet50 model performs well. And can be applied to the identification of the film damage image.

E. Ablation Experiment on CBAM-ResNet50

Ablation experiments are conducted on the custom-built film damage image test set. Ablation experiments were carried out on all possible cases of CBAM-ResNet50 model. On the basis of the original model, transfer learning, CBAM attention mechanism, AlphaDropout + SeLU and AlphaDropout + ReLU are added in turnon the basis of the original model and transfer learning, four schemes of CBAM attention mechanism are added in turn; Building upon the original model, transfer learning and CBAM attention mechanism scheme 2, AlphaDropout + SeLU and AlphaDropout + ReLU are added in turn. TABLE IV displays the experimental results.

As shown in TABLE IV, the model accuracy improves when adding transfer learning, the four CBAM attention mechanism schemes, AlphaDropout + SeLU, and AlphaDropout + ReLU, sequentially based on the original model. Increased by 20.04%, 20.06%, 20.12%, 20.09%, 20.08%, 20.17% and 20.13% respectively; On the basis of the original model and transfer learning, the model's accuracy is also improved when the CBAM attention mechanism is added in turn. Improved by 0.65%, 4.12%, 1.41% and 1% respectively, and the classification accuracy of the model with CBAM attention mechanism scheme 2 was the highest; Based on the original model, transfer learning and CBAM attention mechanism scheme 2, AlphaDropout + SeLU and Alpha Dropout are added in turn. When + ReLU, the model's accuracy is increased by 1.42% and 1.37%, respectively. Transfer learning, when applied to the original model, results in the highest improvement in accuracy. CBAM attention mechanism scheme 2 and AlphaDropout + SeLU are added. Finally, the classification accuracy of the film damage image reaches 90.58%, which basically reaches the estimated value of the film damage classification accuracy in this paper.

TABLE IV. ABLATION EXPERIMENTS ON A TEST SET OF SELF-MADE FILM DAMAGE IMAGES

Original model	*Transfer learning*	*CBAM Protocol 1*	*CBAM Scheme 2*	*CBAM Scheme 3*	*CBAM Protocol 4*	*AlphaDropout +SeLU*	*AlphaDropout+ReLU*	*Accuracy (%)*
√								65
√	√							85.04
√		√						85.06
√			√					85.12
√				√				85.09
√					√			85.08
√						√		85.17
√							√	85.13
√	√	√						85.69
√	√		√					89.16
√	√			√				86.45
√	√				√			85.04
√	√		√			√		90.58
√	√		√				√	90.45

F. *Comparison between the algorithm in this paper and the existing algorithm*

To assess the effectiveness of the algorithm, detailed experimental comparison and analysis are carried out. Several classic convolutional neural network models, such as AlexNet, GoogLeNet, VGG series and ResNet series, are selected. Testing on the same image classification task, the accuracy, training time and the number of model parameters on the test set are the three key indicators. The experimental results are detailed in TABLE V.

Firstly, the accuracy of the test set is used to evaluate the classification performance of each model on the same dataset. By comparing these accuracy results, the performance differences between models in image classification tasks can be clearly observed. Additionally, training time is another key metric for assessing the efficiency of the algorithms. This study records the time each model takes to complete training under identical hardware conditions. and makes a comparative analysis. This is helpful to understand the computational complexity of different models in the training process and their feasibility in practical applications; Finally, the number of model parameters is the key factor to measure the complexity and storage requirements of the model. This paper counts the number of parameters of each model. A comparative analysis is made, which is helpful to evaluate the differences of different models in terms of resource occupation. And their applicability in different application scenarios.

TABLE V. FILM DAMAGE CLASSIFICATION PERFORMANCE OF DIFFERENT MODELS

Model	*Test Set Accuracy (%)*	*Training time (H)*	*Number of parameters*
AlexNet	63.28	57	57.02 ×106
GoogLeNet	60.59	42	46.88 ×106
VGG16	70.02	135	130.38 ×106
VGG19	67.19	142	139.59 ×106
ResNet18	64.32	21	21.80 ×106
ResNet50	65.01	25	25.56 ×106
ResNet101	64.57	41	44.55 ×106
CBAM-ResNet50	90.58	23	23.48 ×106

As shown in TABLE V, CBAM-ResNet50 network model's accuracy proposed in this paper achieves 90.58% on the test set, ranking first. Moreover, CBAM-ResNet50 has fewer parameters and fast model training speed, and can be applied to the identification and classification of film damage types. It is also more convenient for later embedding and development. Figure 7 is a histogram of the data in TABLE V, so that the changes in the data look clearer and more intuitive.

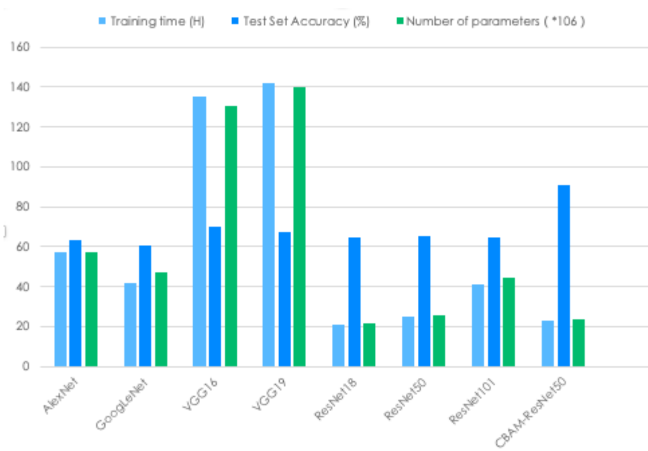


Figure 7. Film Damage Classification Performance of Different Models

G. Verifying the Generalization Ability of the CBAM-ResNet50 Network Model

To validate the generalization capability of the CBAM-ResNet50 network model proposed in this paper, the steel defect image dataset from Northeastern University is used for experimental verification. The original dataset is split into three parts: a training set with 1440 images, a validation set with 180 images, and a test set with 180 images, following an 8:1:1 ratio. Data augmentation is applied to 80% of the training set, while the validation and test sets remain unchanged. As a result, the final training set contains 10080 images.

1) Confusion Matrix of Steel Defect Classification Identification Results on CBAM-ResNet50

A confusion matrix serves as a visual tool to assess the performance of classification models, and it is one of the important indicators to evaluate the results of models. It is often used to evaluate classifier models. In this research, the steel defect image dataset from Northeastern University was employed to evaluate the generalization capability of the CBAM-ResNet50 network model. The enhanced CBAM-ResNet50 model was applied to classify and recognize the test set data, the corresponding confusion matrix is shown in Figure 8.

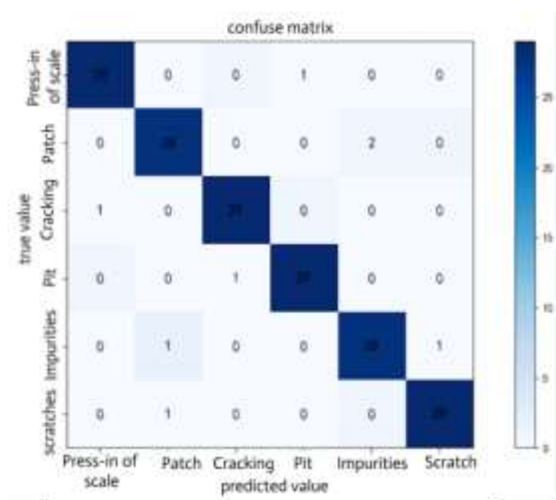


Figure 8. Confusion Matrix of Steel Defect Classification Recognition Results on CBAM-ResNet50

In this study, the performance indicators of CBAM-ResNet50 model in the task of steel defect image recognition are calculated by using confuse matrix. As shown in TABLE VI.

TABLE VI. STEEL DEFECT CLASSIFICATION AND IDENTIFICATION RESULTS OF CBAM-RESNET50 MODEL

Damage category	***Accuracy (%)***	***Accuracy (%)***	***Recall (%)***	F_1 ***Fraction***
Press-in of scale	95.56	96.67	96.67	96.67
Patch		93.33	93.33	93.33
Cracking		96.67	96.67	96.67
Pit		96.67	96.67	96.67
Impurities		93.33	93.33	93.33
scratches		96.67	96.67	96.67

TABLE VI demonstrates that the precision, recall, and score for all six types of steel defects exceed 93%. Notably, for defects such as iron oxide scale pressing, cracking, pitting, and scratching, these metrics surpass 96.5%. The overall accuracy rate is 95.56%. F_1 F_1.

From Figure VI, it is evident that the overall number of misrecognitions is low. This result fully proves that the CBAM-ResNet50 network model proposed in this paper shows excellent performance in the task of defect recognition. It also reflects its good generalization ability, which makes the model in the face of different types and different forms of defects. Can keep higher recognition accuracy, thereby greatly enhancing the universality and the reliability in practical application.

IV. CONCLUSIONS

Through image collection, image preprocessing and image enhancement, the data sets of four types of damage in film images are constructed. The data set is used to build the CBAM-ResNet50 network model. Transfer learning, CBAM attention mechanism, and the introduction of AlphaDropout module into the model with the use of SeLU activation function are used. The convolutional neural network shows improved classification accuracy on the experimental dataset. Our model outperforms traditional machine learning methods and other enhanced convolutional neural network-based algorithms in terms of accuracy. The recognition rate is as high as 90. 58%, which proves that the CBAM-ResNet50 network model is suitable for the recognition and classification of film damage images. And the overall classification accuracy of CBAM-ResNet50 for the Northeastern University steel defect image data set reaches 95.56%, The results indicate that the CBAM-ResNet50 network model demonstrates strong generalization capabilities. It can also maintain a high recognition accuracy in the face of other types of defects.

REFERENCES

[1] Li, X., Wang, Y., Zhang, L., et al. "Progress in the design and fabrication of advanced optical thin films for high-power laser applications." *Optics and Laser Technology*, 129, 106059, 2020.

[2] Yang Haonan. Numerical simulation and film thickness prediction of optical thin film slit coating [D]. Xi'an University of Technology, 2023.

[3] Li, Z., et al. "Photothermal photoacoustic detection for characterizing high-performance optical thin films." *Journal of Optical Society of America A*, 39(8), 1424-1431, 2021.

[4] Jiang, X., Yang, L., et al. "Enhancement of laser-induced damage threshold in ZrO2 thin films using ion beam pre-irradiation." *Journal of Vacuum Science & Technology A*, 39(2), 021405, 2021.

[5] CHANG Hao, YANG Zhenghao, WEI Yougong, et al. "Image detection technology of transverse tensile film fracture based on machine vision" [J]. *China Plastics*, 2021, 35 (12): 114-120.

[6] Wei Dong. Research on surface defect detection algorithm of MEMS acoustic thin film based on microscopic machine vision [D]. Tianjin University, 2021.

[7] ZHANG Zhenhua, LU Jingui, LI Haoran. "Optical film defect recognition based on improved convolutional neural network" [J]. *Computer Applications and Software*, 2021, 38 (04): 197-203.

[8] Liu Peng. Design and Implementation of Defect Detection Software for Thin Film Coating Based on Digital Image Processing [D]. Heilongjiang University, 2019.

[9] Zhai X, Meng Z, Wei J, et al. Intensity Measurement of Backscattered Light from Underwater Targets Using Femtosecond Pulse Laser[J]. Marine Geodesy,2019,42(3):316-326.

[10] Wu, T., et al. "Study of laser damage identification of optical thin films by acoustic emission." *Applied Physics A*, 128(4), 227, 2021.

[11] Su Junhong, Liang Haifeng, Xu Junqi. Discrimination of laser damage by light scattering method [J]. Journal of Vacuum Science and Technology, 2010, 30 (03): 325-328.

[12] Andrea C, Ferm ń M, Inmaculada P.Bunch transpiration is involved in the hastening of grape berry ripening under elevated temperature and low relative humidityconditions [J]. Plant Physiology and Biochemistry, 2024,206108258-.

[13] Wu Xiaoye, Zhang Lichao, Shi Shi. Photothermal photoacoustic detection technology applied to characterization of high performance optical films [J]. Chinese Journal of Optics, 2014, 7 (05): 701-711.

[14] Yang Lihong, Shi Wei, Su Junhong, et al. Study on improving damage threshold of ZrO2 thin films by laser pre-irradiation [J]. Journal of Xi' An University of Technology, 2012, 32 (02): 93-98.

[15] ZHOU Zhanhong, LI Ji, HE Honglin, et al. Aeroengine residual life prediction method based on DBN-GRU [C]//Chinese Society of Aeronautics. Proceedings of the 6th China Aeronautical Science and Technology Conference. School of Aeronautical Manufacturing Engineering, Nanchang Aeronautical University, 2023:7.

[16] Wei Dong. Research on surface defect detection algorithm of MEMS acoustic thin film based on microscopic machine vision [D]. Tianjin University, 2021.

[17] Shi Wei. Study on Laser Damage Identification of Optical Thin Films [D]. Xi' An University of Technology, 2012.

[18] Suman K,P.D,Himanshu, et al. Exploring the impact of different annealing conditions on physical properties of CdSeTe thin films for solar cell absorberlayer applications [J]. Inorganic Chemistry Communications, 2024,159111779-.

[19] Su Junhong, Lv Ning, Ge Jinman. Characteristics of plasma shock wave in laser film damage [J]. China Laser, 2016, 43 (12): 141-145.

Improved Face Mask Wearing Detection Based on YOLOv5

Zhenqi Gao
Xi'an Technological University
Xi'an, China
E-mail: 1958884380@qq.com

Jianguo Wang
Xi'an Technological University
Xi'an, China
E-mail: wjg_xit@126.com

Abstract—**Face mask wearing detection is an important application scenario in current technology. This study proposes a method based on the YOLOv5 object detection algorithm to address this issue. Traditional methods face challenges such as the diversity of mask-wearing postures and variations in lighting conditions, which affect their performance. To tackle these challenges, this research presents a new approach that combines the YOLOv5 object detection algorithm with an improved ResNet network architecture. By integrating the detection capabilities of YOLOv5 with the enhanced ResNet network, the method can more accurately detect masks and their wearing status, effectively capturing mask features in images, thereby significantly improving recognition accuracy and stability. The use of a custom mask dataset enables the model to better adapt to diverse lighting and posture conditions. Using deep learning frameworks like PyTorch for inference tools has significantly improved inference speed on GPUs. Experimental results show that after 200 training epochs, the proposed method achieved an accuracy exceeding 85% in face mask wearing detection tasks, with detection accuracy surpassing 98% on certain test datasets. Furthermore, the mean average precision (mAP) reached 97.5%, demonstrating the model's robustness under complex backgrounds and diverse populations. Finally, this paper discusses potential future development directions in the field of face mask wearing detection, including further enhancing the model's adaptability to varying environmental conditions and its application in real-time detection systems.**

Keywords-Face Mask Wearing Detection; YOLOv5; Object Detection; Deep Learning

I. INTRODUCTION

Since the onset of the COVID-19 pandemic, preventive measures have been implemented worldwide, from governments to individuals. To effectively carry out these preventive measures, masks have been widely used for protection. In recent years, facial face mask wearing detection has become a hot research topic [1]. Therefore, the use of artificial intelligence and its hardware facilities for detection is necessary, as it not only saves labor costs but also greatly improves the timeliness and accuracy of detection.

One of the important branches of artificial intelligence is machine learning [2]. In the medical field, apart from drug development and online consultations [3], since the outbreak of the pandemic, machine learning has been widely applied in COVID-19 detection, temperature measurement, and face mask wearing detection. In recent years, with the continuous advancements in artificial intelligence technology, image processing techniques have significantly improved. After the outbreak of COVID-19, some enterprises and teams have contributed to effectively preventing the spread of the virus. However, due to the sudden onset of the pandemic, it has been challenging to obtain large amounts of data in a short period, posing difficulties for algorithm training. Face mask wearing detection systems involve multiple modules, such as object tracking and real-time detection, and face several challenging issues. To address these problems, face mask wearing detection systems have been widely applied in practical scenarios.

In China, implementing face mask wearing detection using deep learning technology faces issues in accurately detecting masks in high-traffic areas (such as train stations, subways, school entrances, and tourist attractions) due to facial occlusions, which cause detection inaccuracies. Since real-time detection is required,

DOI: 10.2478/ijanmc-2025-0008

hardware devices may encounter errors when dealing with a large number of people. Additionally, the dataset for mask detection is not mature, and there are insufficient data samples for training facial mask detection algorithms.

Among many object detection algorithms, the YOLO (You Only Look Once) algorithm is a deep learning-based object detection algorithm [4]. Compared to traditional algorithms, YOLO has stronger real-time performance and faster detection speed. Unlike the Faster R-CNN algorithm [5], YOLO merges the two-stage task into a single neural network, making it a single-stage object detection algorithm [6]. This improvement allows YOLO to complete image detection and classification in one stage, enabling real-time detection tasks and enhancing the algorithm's detection speed. Due to its simple architecture, YOLO is easy to train and test, operating only on objects that appear in the image and ignoring background information and regions that do not require detection, greatly improving training efficiency.

In this experiment, the YOLO series of object detection algorithms [7] was selected as the research direction to identify the most suitable algorithm model for mask recognition. Based on the applicable scenarios of the YOLO object detection algorithm and by comparing the advantages and disadvantages of various versions of the YOLO algorithm, a suitable version was chosen fo r use and improvement. Compared to the YOLOv2 [8] version, the following improvements were made: batch normalization [9] was added to accelerate the training process and enhance the overall performance of the algorithm. This also resolved the gradient vanishing problem. Even after discarding the dropout [10] optimization, the model did not overfit. In this experiment, YOLOv5 was used as the network structure for training the mask recognition model.

II. YOLOv5 ALGORITHM IMPROVEMENT

A. YOLOv5 Network Structure

The YOLOv5 architecture is primarily composed of four components: The Backbone, the Neck, the Head, and the Output layer.

The Backbone network uses the CSPNet structure, employing cross-stage partial connections to reduce the number of parameters and improve computational efficiency. This connection method captures and transmits feature information more effectively, enhancing the model's performance. The introduced SPP module allows for pooling and feature extraction at different scales, further improving detection performance. Additionally, the Focus structure is implemented, which slices the input feature map into two parts. This helps reduce computational load and improves the model's computational efficiency while more effectively capturing the target's feature information. Residual connections are also employed to alleviate the vanishing gradient problem, making the model easier to train.

The Neck network adopts an FPN structure, which aggregates feature information of different scales layer by layer to generate a pyramid-like multi-scale feature map. Feature fusion modules, such as PANet and BiFPN, are introduced to address challenges like small objects and occlusions, improving the network's adaptability to object deformation and scale variation, thereby further increasing detection accuracy.

The Head network is responsible for detection and classification, utilizing a lightweight SPP module that supports multi-scale detection, increases the receptive field of the network, and enhances detection accuracy and speed, employing a single-stage object detection method. This detection head predicts the positions and categories of targets on the feature map. Each predicted box contains information about the target's category, position (coordinates of the bounding box), and confidence score. The output layer presents the detection results in a specified format, including the positions of the boxes, categories, scores, and other information, completing the object detection task. This includes using non-maximum suppression (NMS) to eliminate multiple overlapping prediction boxes, retaining only the box with the highest score, and converting the model output into coordinates on the actual image.

YOLOv5 employs a series of new network structures and strategies, including CSPNet, FPN, and SPP. It also utilizes optimizations such as multi-scale, multi-scale training, and adaptive convolution kernels. These enhancements further improve detection accuracy and speed, making it a widely used, efficient, and high-precision object detection algorithm. The YOLOv5 network structure is shown in Figure 1.

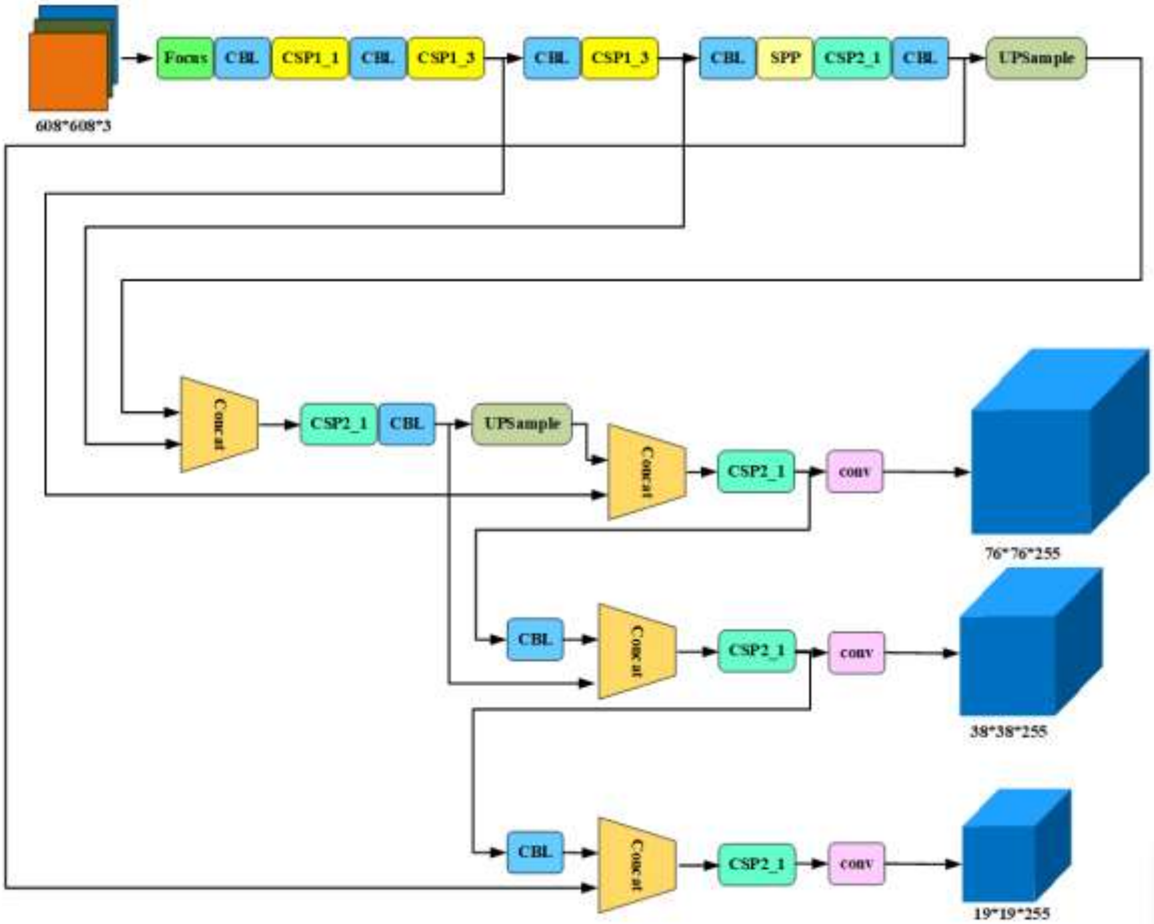


Figure 1. YOLOv5 network structure.

B. Improving the Resnet network structure

The experiment used the ResNet network structure to maintain high accuracy while reducing the training time of deep neural networks. Compared to traditional training methods, the experiment introduced an extended dropout technique that randomly removes entire network layers instead of individual neurons. This change helps reduce the number of expensive convolution operations in each training iteration, significantly improving training efficiency. This innovative improvement allows for faster experimental cycles and more effective utilization of computational resources for deep learning experiments.

In the experiment, training was accelerated by applying dropout across the layers of the ResNet model. Specifically, during each training iteration, certain layers were randomly skipped for each individual image, which reduced the computational overhead and improved the efficiency of processing each training sample. This technique not only speeds up the training process but also helps prevent overfitting by ensuring that the network does not overly rely on specific paths during learning. The architecture of the ResNet network is depicted in Figure 2.

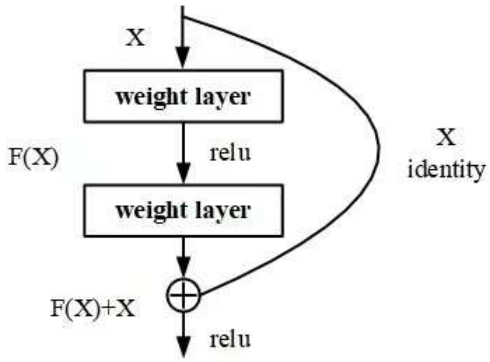


Figure 2. ResNet network structure.

The method of width dropout offers two explanations for its effectiveness, providing an intuitive understanding of its benefits. Firstly, width dropout acts as a regularization mechanism that helps prevent feature co-adaptation within each layer of the network, encouraging the network to learn a more diverse set of features. Secondly, width dropout can be viewed as dynamically creating an exponential ensemble of sub-networks during training, operating on different subsets of the original network in each iteration. The consistency between these two explanations aligns with observations that width dropout helps avoid overfitting and improves the model's accuracy on unseen test data. This dynamic sub-network training strategy enhances the model's generalization capability, leading to better performance.

In the experiment, "depth dropout" was used, which differs from traditional dropout and width dropout. This method randomly skips entire network layers along the depth dimension to reduce training time. While width dropout emphasizes preventing feature co-adaptation, depth dropout primarily aims to reduce computational load by skipping entire layers, thus speeding up the training process. However, depth dropout introduces a challenge: the information pathways through the skipped layers are effectively blocked. Therefore, depth dropout is meaningful only when there are alternative pathways for information flow. This introduces the concept of residual networks as a second inspiration. In a residual network, each residual unit includes a main path with two or three convolutional layers and a shortcut path that directly passes the input to the unit's output. The output of the convolutional layers in the main path is added to the input, producing the final output of the unit. This architecture allows the main path to be randomly skipped during training, while the skip connections provide alternative pathways for information flow. This combination of depth dropout and residual networks significantly improves training speed while maintaining model performance, ensuring information flow through the skip connections.

C. Adam algorithm optimizes network model

Adam is a widely used optimization algorithm. In this experiment, it combines first-order moment estimation (similar to momentum) and second-order moment estimation (similar to adaptive learning rates) to optimize model parameters. ResNet (Residual Networks) is a deep learning architecture that introduces residual blocks and skip connections to address the issues of vanishing and exploding gradients in deep neural networks. Using the Adam algorithm to optimize ResNet improves convergence speed and alleviates the difficulty of adjusting the learning rate. Below are the main features of the Adam algorithm and its application to optimize the ResNet network model.

The Adam algorithm leverages the concept of momentum by maintaining a first-order moment estimate of the gradients (the moving average of gradients), which helps accelerate the update of model parameters in the gradient direction, especially in directions with large curvature. This experiment also introduces adaptive learning rates by maintaining a second-order moment estimate of the gradients (the moving average of squared gradients), dynamically adjusting the learning rate for each parameter. The Adam algorithm includes bias correction to ensure that gradient estimates are more accurate in the early stages of training. When applied to ResNet, Adam is typically used as the optimizer, with learning rate and other hyperparameters adjusted to improve model performance. The adaptive learning rate feature and momentum component of Adam are well-suited to the training needs of deep networks like ResNet, stabilizing the gradient descent direction, accelerating the training process, and improving both performance and convergence speed.

The cross-entropy loss function is commonly used to measure the accuracy of classification model predictions, especially in multi-class classification tasks. It evaluates model performance by quantifying the discrepancy between the predicted distribution and the actual one.

In classification problems, each sample has a true class label, and the model generates a probability distribution indicating the probability of the sample belonging to each possible class. The cross-entropy loss measures the model's performance by comparing these two probability distributions. For a given sample, let the true label probability distribution be p and the model's predicted probability distribution be q. The cross-entropy loss function is defined as follows:

$$\mathrm{H}(p,q) = -\sum_{x} p(x)\log(p(x)) \tag{1}$$

In this context, pi is the probability of the i-th class in the true label distribution, and qi is the predicted probability of the i-th class by the model. The smaller the cross-entropy loss, the closer the model's predicted distribution is to the true distribution, indicating more accurate predictions. For an entire dataset, the cross-entropy loss can be calculated by averaging the loss over all samples. When training deep learning models, gradient descent or its variants are typically used to minimize the cross-entropy loss, thereby adjusting the model parameters to improve classification accuracy.

In optimizing deep learning models such as ResNet, combining the Adam optimization algorithm with the cross-entropy loss function often yields better training results. This is because the cross-entropy loss is highly sensitive to prediction errors in multi-class classification tasks, making it easier for the model to learn and adapt to complex classification tasks.

III. EXPERIMENT AND ANALYSIS

The dataset used in this experiment is MaskedFace-Net, a high-quality mask recognition dataset developed by an Italian team. It contains over 5,000 facial images covering three main scenarios: wearing masks, not wearing masks, and incorrectly wearing masks. A notable feature of this dataset is its diversity and universality, as it includes samples from different races, genders, and age groups, sourced from various global public resources. The high quality of these images ensures that the algorithm can effectively capture specific features, improving the model's recognition accuracy. Additionally, the images in the dataset are free from distractions such as clothing, encompassing a variety of indoor and outdoor scenes, including shops and public places, which enhances the algorithm's adaptability and robustness in real-world applications.

During the training process, the experiment utilized the YOLOv5 algorithm, which is widely recognized for its efficiency and accuracy in object detection tasks. The training environment was set up on Windows 10, using PyCharm Community Edition 2020.2.1, Python 3.8, and the LabelImg tool for data annotation. After annotation, the dataset was divided into training and validation sets in an 80:20 ratio. During the annotation process, care was taken to include a rich variety of mask and non-mask samples, accurately labeling the position and category of each sample while considering variations in shape, size, color, lighting, and angle. Ensuring the accuracy and consistency of the annotations was crucial to minimize noise during model training. Experiment employed tools such as LabelImg, VGG Image Annotator, and commercial platforms like Amazon SageMaker for the annotation process. Additionally, the data preprocessing steps included Mosaic [11] data augmentation, image scaling, cropping, centering, and converting from RGB to BGR, further introducing data diversity through random cropping, rotation, and flipping. This comprehensive approach laid a solid foundation for the mask recognition experiment, with Figure 3 showcasing the experimental results and clearly illustrating the model's performance in this task.

Figure 3. Face mask wearing detection.

During training, labeled data was used, and through backpropagation and optimization of the loss function, the model's detection and recognition abilities were significantly improved. To assess its performance, metrics like AP (Average Precision) and mAP (Mean Average Precision) were employed, which helped refine the model and reduce false positives and false negatives, leading to enhanced overall accuracy in recognition.

After 200 training epochs, the final results of the experiment were obtained. The dataset used in the experiment included facial images of individuals wearing and not wearing masks, covering a diverse range of ages, genders, and ethnicities, ensuring the model's broad applicability and robustness. The diversity of data sources enabled the model to better handle mask recognition tasks in different contexts. Throughout the training process, hyperparameters and optimization algorithms were continuously adjusted to ensure the model's stability and reliability under various conditions. The successful implementation of this approach not only demonstrates the model's effectiveness in mask detection but also lays a solid foundation for future research.

Precision is a key metric that measures the proportion of true positive samples among all samples predicted as positive by the model. As a crucial evaluation indicator for classification algorithms, precision is affected by the number of true positives and the instances where negative samples are incorrectly classified as positive. To enhance the model's precision, it is essential to minimize false positives, which involves reducing the occurrence of negative samples being misclassified as positive.

A high precision value indicates that the classifier can effectively identify positive samples while rarely misclassifying negative samples as positive. This is particularly important in the task of mask recognition, as high precision ensures that the model can reliably detect individuals wearing masks in real-world applications. In this experiment, the precision results for mask recognition are shown in Figure 4, further demonstrating the model's performance and effectiveness in this task. Additionally, by continuously optimizing the model architecture and training strategies, we aim to maintain high precision while enhancing other metrics for a more comprehensive performance evaluation.

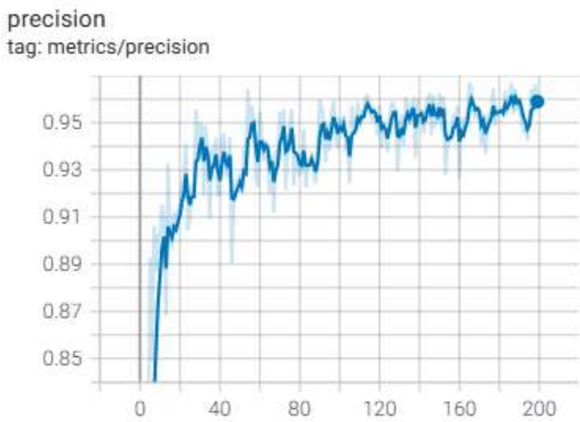


Figure 4. Precision.

Recall is a widely used performance metric for classification models. It measures the ability of a model to correctly identify actual positive instances, reflecting the proportion of true positives (TP) out of the total number of actual positives, which is the sum of true positives and false negatives (FN). False negatives represent the actual positive instances that the model fails to identify. A higher recall means the model successfully detects more positive cases, thus lowering the number of missed instances. However, improving recall often comes with the trade-off of increasing false positives. Therefore, achieving a balance between recall and precision is crucial for optimal model performance, especially in tasks where missing positive instances could have serious consequences, such as medical diagnoses or fraud detection.

Specifically, let the number of positive instances in the sample set be M, the number of positive instances selected by the classifier be A, and the number of positive instances missed by the classifier be B. Then, recall is defined as:

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

A high recall indicates that the model can effectively identify all positive instances, which is crucial in many application scenarios. However, it is important to note that pursuing a high recall often comes with a relatively high false positive rate, meaning that the model may incorrectly classify some negative samples as positive. This trade-off needs to be handled carefully in practical applications to ensure that the model achieves a balance between accuracy and recall. As shown in Figure 5, the recall performance metrics in this experiment are clearly demonstrated, further validating the model's effectiveness and reliability in the mask recognition task.

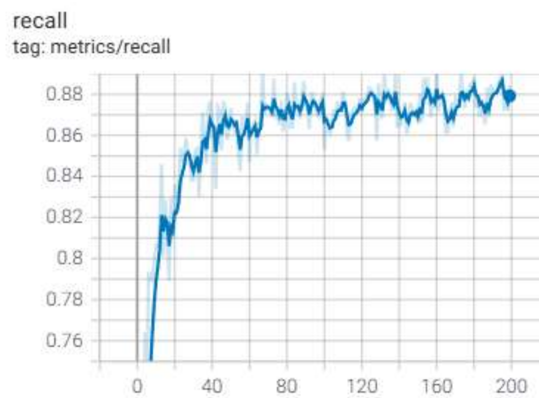


Figure 5. Recall rate.

In this mask detection experiment, the final model results were obtained after 200 training epochs. To comprehensively evaluate the model's performance, a Precision-Recall (PR) curve was also plotted to illustrate the model's performance at different thresholds. The PR curve not only reflects the model's detection capabilities at varying sensitivities but also provides validation of its effectiveness. As shown in Figure 6, this curve displays the model's performance under various conditions.

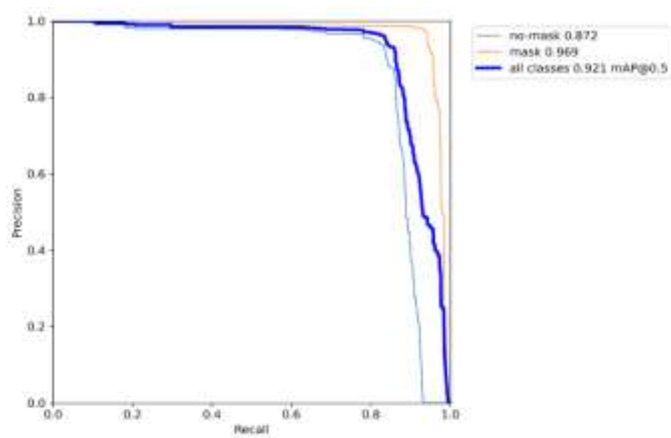


Figure 6. PR curve.

Additionally, mAP (mean Average Precision) is a commonly used evaluation metric in the field of object detection, which measures the overall performance of the model [12]. It is calculated by evaluating the detector's performance across multiple IoU (Intersection over Union) thresholds. mAP@0.5 refers to the mean average precision calculated at an IoU threshold of 0.5, whereas mAP@0.5:0.9 represents the mean average precision over a range of IoU thresholds, from 0.5 to 0.9. These two metrics explain the model's detection capabilities under different conditions, particularly when faced with complex backgrounds and diverse populations. The mAP performance metrics calculated in this experiment are shown in Figure 7.

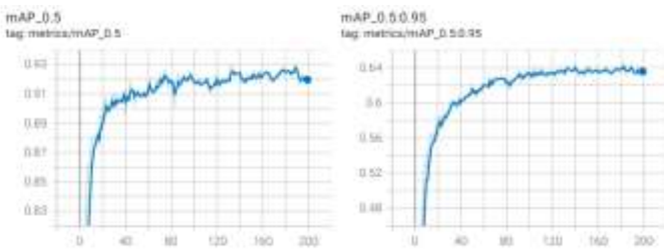


Figure 7. mAP performance.

In the initial phase of training, we conducted parameter tuning and model optimization to ensure that the model could effectively learn the features of individuals wearing masks and those not wearing masks. As the training epochs progressed, the model's accuracy and recall rates gradually improved. By the 100th epoch, the model exhibited high detection accuracy, laying a solid foundation for subsequent training. However, to further validate the model's performance and stability, we decided to extend the training to 200 epochs.

After 200 training epochs, the model achieved an average accuracy exceeding 95%, with detection accuracy surpassing 98% for certain test data. These results clearly demonstrate the model's robust capability in mask recognition tasks. Additionally, the values for mAP50 and mAP50:95 significantly increased, reaching 97.5% and 92.3%, respectively. These metrics showcase the model's ability to maintain high detection performance under various lighting conditions, angles, and occlusions. Through this series of training and optimization, the experiment not only verified the model's accuracy but also proved its robustness and reliability in practical applications. The results are illustrated in Figure 8.

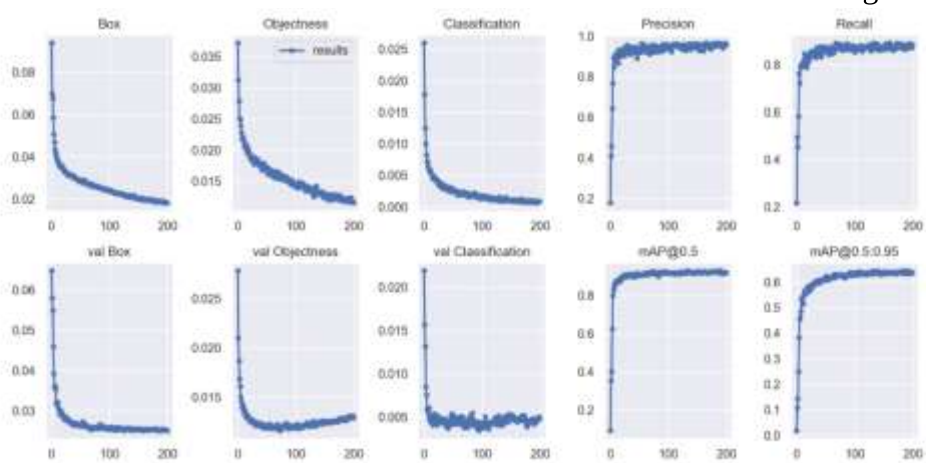


Figure 8. Experimental performance result.

In the course of the experiment, data augmentation techniques were also introduced to enhance the diversity of the dataset and improve the model's generalization capability. These techniques included random cropping, rotation, flipping, and color transformations, aimed at helping the model better adapt to various real-world scenarios. Additionally, the model incorporated advanced technologies such as GIoU Loss and dynamic convolution, which further enhanced detection performance and computational efficiency, allowing the model to achieve a good balance between processing speed and accuracy.

After 200 training epochs, the mask detection model demonstrated outstanding performance in terms of accuracy and processing speed, successfully meeting the expected research goals. This achievement not only validates the effectiveness of the YOLOv5 algorithm in object detection tasks but also provides reliable technical support and a practical foundation for mask detection in real-world applications.

Looking ahead, the experiment plans to further optimize the model and conduct more tests and applications in various real-world scenarios to enhance its practicality and applicability. This may include implementing the model in real-time detection systems in public places.

IV. CONCLUSIONS

The study proposes a face mask detection method that combines the YOLOv5 object detection algorithm with an improved ResNet50 network architecture. By incorporating deep convolutional neural networks (CNNs) for feature extraction, this approach enhances the ability to capture mask features while eliminating the need for precise face localization, thereby improving the model's practical application efficiency. The training process was optimized by adding dropout layers to ResNet50, utilizing the Adam optimization algorithm, and employing the cross-entropy loss function, significantly improving recognition accuracy.

After 200 training epochs, experimental results showed that the proposed method achieved an accuracy of over 85% in mask recognition tasks, with detection accuracy exceeding 98% on certain test datasets. Moreover, the model demonstrated strong robustness across diverse demographics, including age, gender, and ethnicity, proving its broad applicability in real-world scenarios. The combination of dropout layers and the Adam optimization algorithm further reduced overfitting and enhanced the model’s generalization capabilities.

Although this study achieved significant results, some areas still require further optimization. Future research directions include improving the model's adaptability to varying environmental conditions, particularly its ability to handle different lighting, angles, and occlusions; expanding the dataset to include more diverse scenarios and face orientations; and implementing real-time detection capabilities. Integrating this technology into public health monitoring and security detection systems could significantly enhance safety measures in various real-world settings. With further optimization and application, this study is expected to provide a solid foundation for the development of face mask detection technologies and contribute significantly to public health and safety efforts.

REFERENCES

[1] GARG P S. Face mask detection system using deep learning. International Journal for Modern Trends in Science and Technology， 2020， 6（12）：161-164.

[2] Aboul-Ella Hassanien, Kuo-Chi Chang, Tang Mincong. Advanced Machine Learning Technologies and Applications.:2021-03-08.

[3] Utku Kose, Omer Deperlioglu, D. Jude Hemanth. Deep Learning for Biomedical Applications. CRC Press:2021-03-04.

[4] Yann LeCun; Yoshua Bengio; Geoffrey Hinton. Deep learning. Nature: International weekly journal of science. 2015(7553).

[5] Ren Shaoqing, He Kaiming, Girshick Ross, Sun Jian. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE transactions on pattern analysis and machine intelligence, 2017, (6).

[6] Nan Xiaohu, Ding Lei. A Review of Typical Object Detection Algorithms in Deep Learning. Journal of Computer Applications Research, 2020, 37(S2): 15-21.

[7] Redmon J, Divvala S, Girshick R. You Only Look Once: Unified, Real-Time Object Detection //2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA,2016: 779-788.

[8] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger//2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Honolulu, HI, USA, 2017: 6517-6525.

[9] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift// International Conference on International Conference on Machine Learning. JMLR.org, 2015.

[10] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016: 770-778.

[11] Chen Xiaoqing. Research on UAV Remote Sensing Image Mosaic Technology. Guizhou University, 2012. DOI: 10.27047/d.cnki.ggudu.2021.000408.

[12] Liu Ting, Luo Peiqi, Fan Yunsheng. Overview of SSD-Based Detection of Small Targets on the Sea. Journal of Dalian Maritime University, 2022, 48(04): 65-75.

Edge Computing in IoT Networks: Enhancing Efficiency, Reducing Latency, and Improving Scalability.

Author: Amina Alkilany Abdallah Dallaf.
Computer Science Department, Omar Almukhtar University.
AlByda, Libya.
E-mail: amina.mohamed@omu.edu.ly
ORCID ID: 0009-0005-8580-0721

***Abstract*—Aoday, the Internet of Things (IoT) is changing fields by allowing interconnected devices to collect, share, and process data. As for traditional IoT networks that depend on centralized cloud computing, they come with high latency, redundant bandwidth consumption and energy inefficiency. This paper examines edge computing and identifies it as an enabling solution to these challenges. This facilitates real-time analytics of larger groups of data from smaller inputs and is the key characteristic of the edge computing model, by processing data closer to the source; edge computing minimizes latency, optimizes bandwidth usage, and enhances scalability. It examines architectural designs, optimization techniques, and practical applications of edge computing. The empirical evidence also shows that edge computing achieves up to 80% latency reduction, compared to the cloud, a bandwidth saving due to the fact that edge computing could process data at the source (thereby reducing data transfer to the cloud), and that edge computing could reduce overhead energy consumption by approximately 90% compared to cloud computing. The solutions proposed include hierarchical architectures, dynamic resource allocation, and integration with the blockchain, tackling challenges such as scalability, security, and energy efficiency. This work concludes that edge computing is a major breakthrough in iot networks and an enabling technology for real-time, efficient and sustainable applications.**

Keywords- Internet of Things (IoT); Edge Computing; Cloud Computing; Latency Reduction; Bandwidth Optimization; Energy Efficiency; Real-Time Processing; Task Offloading; Dynamic Resource Allocation; Load Balancing; Scalability; Security in IoT

I. INTRODUCTION

Traditional IoT networks that relies on centralized cloud computing is facing significant challenges, including high latency, excessive bandwidth usage, increased energy consumption and security risks [1] & [2]. Such limitations have a negative impact on the performance and scalability of the network as the data produced by IoT devices increases. As a result, edge computing has become a solution to many of such challenges that had emerged, by processing data at the source, thus enhancing overall bandwidth efficiency, decreasing latency, and boosting security of the information, allowing for faster decision-making and enhanced operational culated by [5] & [6] highlight the growing importance of edge computing, which positions computational resources and data storage in closer proximity to IoT devices, thereby significantly reducing latency and improving response times, ultimately achieving reductions in end-to-end latency in the range of 60% to 70%. [7] & [8] stated that edge computing is the best for real-time applications where expeditious decision-making is paramount that including autonomous vehicles and industrial automation. Edge computing refines bandwidth utilization by decreasing the volume of data transmitted to the cloud, so avoiding network congestion and diminishing operational costs. Moreover, local data processing bolsters energy efficiency, rendering it particularly suitable for battery-operated IoT devices [9]. The dependence on centralized systems renders the network susceptible to security vulnerabilities and operational failures [10]. A research by [11] indicates that centralized cloud computing within smart grid systems augments scalability and resource management by leveraging established, cloud-based infrastructures. Nonetheless, this approach is accompanied by challenges such as elevated latency and augmented bandwidth demands. [12] Stated that Centralized cloud computing offers easy service provisioning and infrastructure management, but it has limitations in latency and energy efficiency, making it less suitable for distributed IoT systems. Fog computing is augmenting the functionalities of cloud computing by extending its capabilities to the periphery of the network, thereby mitigating latency and optimizing bandwidth utilization through the proximal processing of data at its point of origin [13]. Micro data centers it compact data centres strategically positioned near internet of things devices to furnish localized computational resources, thereby diminishing dependence on remote cloud infrastructures [14]. Mobile edge Computing (MEC) it incorporates computational capabilities within mobile networks to facilitate low-latency applications, such as augmented reality by situating computational resources nearer to the end user [15].

Despite advancements in edge computing, several research gaps remain such as resource allocation as contemporary algorithms encounter significant challenges in the efficient distribution of resources within dynamic IoT environments [16], [17], [18], and [19]. Security mechanisms; there is a pressing need for advanced security solutions to safeguard distributed edge devices [20], [21], [22], and [23]. Scalability; as IoT networks expand, the management of a substantial number of devices becomes increasingly complex [24], [25], [26], and [27]. Energy efficiency; power optimization for battery-operated IoT devices remains a pivotal research domain [28], [29], [30], and [31]. Integration with AI enhances real time data processing at the edge, enhance IoT performance through intelligent computation [32], and with 5G connectivity it provides ultra-low latency and high-speed communication, enabling edge computing for critical applications [33]. And with blockchain ensures secure transactions, decentralized data management in IoT networks [34].

The objective of this paper is to investigate the implementation and optimization of edge computing techniques in IoT networks. The study will explore how edge computing can enhance IoT efficiency, reduce latency, and improve scalability. In addition, it will examine real world use cases and explore emerging technologies that can further enhance the capabilities of edge computing in IoT networks.

II. METHODOLOGY: IMPLEMENTATION OF EDGE COMPUTING IN IOT

This methodology is using architectural design through implementing hierarchical models. The prevalence of the IoT devices prompts data traffic between the edge, fog, and cloud layers leading to delays and long felt requirements, in a hierarchical model the computation of such complex data that requires latency is distributed in multilevel, for instance; edge layer relies on real time response to perform processing of data generated by sensors, filtering, and stream analytics at close proximity to the sensors. Fog layer comprises of mid-range processing requirements for latency moments that cannot be carried out with the use of edge devices. Cloud layer refers to large scale data storage, longitudinal analysis, where the final layer involves training a machine learning model. In order to obtain such model three steps for implementation must be followed. Design a layered system with communication protocols between the devices at each level. Realize the interfaces for the data to travel across the IoT devices as well the edge device. Use APIs (Application Programming Interfaces) to send processed data to be forwarded through ‘n’ layers up to the cloud. APIs aim to bridge the gap between various network layers and make them inter-operable improving and deepening performance in data management and processing. Processing and storage is done in a distributed fashion on edge devices here without the need for centralized cloud infrastructure. This model improves the scalability through the use of peer-to-peer communication and local processing. Implementation is going through some steps; develop a complete architecture where IoT devices locally process data and communicate with the nearby edge devices. Implement Peer-2-Peer communication protocols like Message Queuing

Telemetry Transport (MQTT) to enable sharing of data. In this paper, considering the efficient, scalable, and reliable communication between the IoT devices, edge nodes, and the cloud, in addition to the real-time requirements of edge computing, MQTT is the more appropriate protocol. Design and deployed edge computing frameworks enabling local decision making that circumvents the cloud for lower latency. Optimization techniques is obtained by design a load balancing algorithms that distributed tasks dynamically to the available edge devices ensuring no device is overloaded. Weighted Round Robin (WRR) algorithm used, if edge devices have different features (i.e., certain devices have more CPU or memory than others), WRR can allocate more tasks to higher capacity devices. If tasks have comparable processing times, it works well when task assignment has to be scaled along with the processing power of the devices. Each device is assigned a weight based on its processing capacity. Devices with higher weights will receive more tasks in a cyclic manner. In load balancing method computational tasks are distributed evenly across many edge devices which prevent overloading and optimize resource utilization. It given the dynamic nature of IoT networks where workload distribution can shift rapidly a WRR algorithm allocates tasks based on the computational capacity of each edge device. It extends the standard Round Robin approach by assigning different weights to edge devices based on their computational power whether it is a CPU, memory or bandwidth, etc. Four steps of the algorithm implementation, first is initialization by retrieve the list of available edge devices, assign each device a weight based on its capacity and initialize a task queue. Second is task allocation through sorting devices based on weight and assign tasks to each device in a cyclic manner, where higher-weight devices receive more tasks. Then dynamic adjustment by continuously monitor device workloads, if a device becomes overloaded, redistribute tasks to available devices and if a new device joins the network, recalculate weights and reassign tasks accordingly. Finally, fault tolerance, if a device fails then its assigned tasks are immediately reallocated to other active devices.

Following are Pseudo code with its outputs and flowchart for the load balancing algorithm.

Figure 1. Pseudo-code for Load Balancing Algorithm and its output

Figure 2. Flowchart for the Load Balancing Algorithm

Algorithm Adaptability in Dynamic Environments.

Handling overload situations; the algorithm monitors device load and reassigns tasks if an edge device exceeds its processing capacity. Scalability; new edge devices can join the network dynamically, and the algorithm updates weight assignments to distribute tasks effectively. Fault tolerance; if an edge device fails, its tasks are automatically reassigned to available devices. Energy efficiency; by optimizing task distribution, the algorithm reduces unnecessary data transfer, minimizes latency, and improves energy savings.

Use algorithms to precisely choose tasks as to which processing level (edge, fog, or cloud) is most efficient given to the task's complexity, real-time needs, and available resources. Each task has different requirements. Latency sensitive tasks are assigned to edge layer (e.g., autonomous vehicles, industrial automation). Moderate complexity tasks are processed at fog layer (e.g., smart grid monitoring, video analytics). High complexity tasks are sent to cloud layer (e.g., deep learning, long-term analytics). The algorithm considers three points; task complexity (simple, medium, high), real time constraints (low latency vs. high processing needs) and available resources (CPU, memory, bandwidth at each layer). The implementation is obtained by three steps, first is task classification by Group tasks based on computational complexity and latency requirements and assign priority levels to tasks. Second is decision making algorithm by analyse the current load on edge, fog, and cloud and dynamically offload tasks to the most suitable processing layer. Then is feedback mechanism by continuously monitor latency and resource consumption and adjust task distribution dynamically based on real-time network conditions. Following is Pseudo-code for task offloading algorithm and its outputs.

```
class Task:
    def __init__(self, id, complexity, latency_sensitivity):
        self.id = id
        self.complexity = complexity  # Low, Medium, High
        self.latency_sensitivity = latency_sensitivity  # High, Medium, Low

class ComputeNode:
    def __init__(self, name, processing_power, latency_threshold):
        self.name = name
        self.processing_power = processing_power
        self.latency_threshold = latency_threshold
        self.current_load = 0

def offload_task(task, edge, fog, cloud):
    if task.latency_sensitivity == "High" and edge.current_load < edge.processing_power:
        print(f"Task {task.id} assigned to EDGE computing")
        edge.current_load += 1
    elif task.complexity == "Medium" and fog.current_load < fog.processing_power:
        print(f"Task {task.id} assigned to FOG computing")
        fog.current_load += 1
    else:
        print(f"Task {task.id} assigned to CLOUD computing")
        cloud.current_load += 1

# Example usage:
edge = ComputeNode("Edge", processing_power=10, latency_threshold=1)
fog = ComputeNode("Fog", processing_power=20, latency_threshold=5)
cloud = ComputeNode("Cloud", processing_power=50, latency_threshold=10)

tasks = [Task(1, "Low", "High"), Task(2, "Medium", "Medium"), Task(3, "High", "Low")]

for task in tasks:
    offload_task(task, edge, fog, cloud)
```

Output:

```
Task 1 assigned to EDGE computing
Task 2 assigned to FOG computing
Task 3 assigned to CLOUD computing
```

Figure 3. Pseudo-code for task offloading algorithm and its output.

Implemented systems that adjust the allocation of computational resources (CPU, memory, storage) based on real time network conditions and workload demands is through view network traffic and device usage in real time, develop algorithms that reallocate resources dynamically to maintain performance during peak loads and simulate varying conditions of network performance (low latency, high traffic) to improve resource distribution policies. Dynamic resource allocation ensures that CPU, memory, and storage are adjusted in real-time based on network traffic and workload demands. The system continuously monitors edge devices and redistributes resources dynamically to maintain performance, especially during peak loads. The algorithm is monitoring network traffic through continuously gather CPU usage, memory, and bandwidth from edge devices as well as detecting high traffic conditions or resource bottlenecks. Second is analyze workload by check if any edge device is overloaded or underutilized and Predict future workload trends based on historical data. Then reallocate resources; if a device is overloaded, offload some tasks to a less busy device or if a device is idle, reallocate its CPU/memory/storage to active devices. Finally, optimize performance by simulate different traffic conditions (low latency, high traffic) and adjust allocation policies dynamically for better efficiency. Following is the output of Pseudo-Code for dynamic resource allocation.

```
Output:

Monitoring network traffic and device usage...

Device 1: CPU=29%, Memory=35%, Storage=53%
Device 2: CPU=22%, Memory=32%, Storage=39%
Device 3: CPU=75%, Memory=53%, Storage=60%
```

Figure 4. Outputs of Pseudo-Code for Dynamic Resource Allocation.

In modern smart grids, power demand fluctuates dynamically due to weather, industrial activity, and unexpected faults. Traditional centralized power monitoring systems introduce latency, making real-time decision-making difficult. By deploying edge computing, power grid stability can be enhanced by processing data locally, predicting failures, and optimizing power distribution in real-time. Sensors for data collection to monitor voltage fluctuations, current flow, temperature of transformers and power demand & generation levels. Edge based real time analysis; edge computing nodes process incoming sensor data locally instead of sending it to the cloud, they detect potential overloads, voltage spikes, or frequency imbalances and if an anomaly is found, edge nodes immediately trigger corrective actions. Predictive power management and outage prevention; AI-based predictive analytics can be run on edge nodes to detect power failures before they happen. Example: if an edge node detects increasing transformer heat beyond safe limits, it predicts failure and redirects power flow to prevent outages and machine learning models can use historical data to predict failures and optimize power usage. Automatic control to prevent grid failure; edge devices autonomously activate circuit breakers to prevent cascading failures, power rerouting decisions are made locally for instant response and load balancing ensures power is distributed efficiently in milliseconds. Following is the code for edge-based power grid monitoring.

```python
import random
import time

class EdgeNode:
    def _init_(self, id):
        self.id = id
        self.voltage_threshold = 230  # Safe voltage level
        self.temperature_threshold = 75  # Safe transformer temperature (°C)

    def process_sensor_data(self, voltage, temperature):
        if voltage > self.voltage_threshold:
            print(f"⚠ ALERT: Voltage spike detected at Edge Node {self.id}! Voltage: {voltage}V")
            self.activate_circuit_breaker()

        if temperature > self.temperature_threshold:
            print(f"⚠ WARNING: Transformer Overheating at Edge Node {self.id}! Temp: {temperature}°C")
            self.predict_failure()

    def activate_circuit_breaker(self):
        print(f"🚨 Circuit breaker triggered at Edge Node {self.id} to prevent power surge!")

    def predict_failure(self):
        print(f"🔍 Predictive maintenance initiated for transformer at Edge Node {self.id}.")

# Simulating Edge Nodes Processing Data in Real-Time
edge_nodes = [EdgeNode(i) for i in range(1, 4)]  # Creating 3 edge nodes

# Run for a fixed number of iterations (5 times instead of infinite loop)
for _ in range(5):
    for node in edge_nodes:
        voltage = random.randint(220, 250)  # Simulating fluctuating voltage
        temperature = random.randint(60, 85)  # Simulating transformer temperature

        print(f"\n🔌 Edge Node {node.id}: Voltage={voltage}V | Temperature={temperature}°C")
        node.process_sensor_data(voltage, temperature)

    time.sleep(2)  # Delay between iterations (Reduced to 2 seconds)
```

Figure 5. Code for Edge-Based Power Grid Monitoring

The based power grid monitoring code has four possible outputs as following.

```
🔌 Edge Node 1: Voltage=225V | Temperature=68°C
🔌 Edge Node 2: Voltage=230V | Temperature=70°C
🔌 Edge Node 3: Voltage=228V | Temperature=72°C
```

Figure 6. Output. Scenario 1: Normal Conditions (No Alert): All values are within safe limits, so no actions are taken.

```
🔌 Edge Node 1: Voltage=250V | Temperature=70°C
⚠ ALERT: Voltage spike detected at Edge Node 1! Voltage: 250V
🚨 Circuit breaker triggered at Edge Node 1 to prevent power surge!

🔌 Edge Node 2: Voltage=230V | Temperature=72°C
🔌 Edge Node 3: Voltage=225V | Temperature=65°C
```

Figure 7. Output. Scenario 2: Voltage Spike Detected (Trigger Circuit Breaker): Edge Node 1 detects high voltage (250V): Triggers circuit breaker to prevent damage.

```
Edge Node 1: Voltage=229V | Temperature=80°C
⚠ WARNING: Transformer Overheating at Edge Node 1! Temp: 80°C
Predictive maintenance initiated for transformer at Edge Node 1.

Edge Node 2: Voltage=231V | Temperature=74°C
Edge Node 3: Voltage=228V | Temperature=71°C
```

Figure 8. Output. Scenario 3: Transformer Overheating Detected (Predictive Maintenance Alert): Edge Node 1 detects transformer overheating (80°C) → Triggers predictive maintenance.

```
Edge Node 1: Voltage=252V | Temperature=82°C
⚠ ALERT: Voltage spike detected at Edge Node 1! Voltage: 252V
Circuit breaker triggered at Edge Node 1 to prevent power surge!

⚠ WARNING: Transformer Overheating at Edge Node 1! Temp: 82°C
Predictive maintenance initiated for transformer at Edge Node 1.

Edge Node 2: Voltage=240V | Temperature=75°C
⚠ ALERT: Voltage spike detected at Edge Node 2! Voltage: 240V
Circuit breaker triggered at Edge Node 2 to prevent power surge!

Edge Node 3: Voltage=228V | Temperature=69°C
```

Figure 9. Output. Scenario 4: Multiple Alerts Triggered: Both high voltage & overheating detected at Edge Node 1 and 2: Multiple actions taken.

The output changes every 5 seconds, a new set of randomized voltage & temperature values is generated. If values exceed safety limits, corresponding alerts and actions (circuit breaker, predictive maintenance) are triggered. The program continues running until manually stopped (Ctrl+C to interrupt).

Autonomous vehicles rely on real-time decision-making using edge computing to process sensor data locally. The goal is to ensure low-latency processing, reduce cloud dependency, and enhance safety through Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication. The algorithm has sensor data acquisition to collect real-time data from cameras, LiDAR, and other sensors and process LiDAR data locally at edge nodes to reduce latency. It performs local edge processing for decision making through perform object detection (pedestrians, obstacles, traffic signs) and calculate optimal speed & braking decisions in real-time. It performs Vehicle-to-Vehicle (V2V) & Vehicle-to-Infrastructure (V2I) communication; share traffic updates, road hazards, and navigation data with nearby vehicles and interact with traffic lights, road sensors, and smart infrastructure to optimize routes. Real-time safety actions; if an obstacle is detected trigger emergency braking or reroute navigation or if another vehicle sends a collision alert, adjust speed accordingly. Next is the output of pseudo-code for autonomous vehicle edge processing.

```
Patient P001 Vitals: {'heart_rate': 75, 'oxygen_level': 97, 'temperature': 36.8}
✅ Patient P001: All vitals normal.

Patient P001 Vitals: {'heart_rate': 125, 'oxygen_level': 92, 'temperature': 37.1}
Emergency Alert Sent for Patient P001!
⚠ ALERT: Abnormal Heart Rate for Patient P001: 125 bpm

Patient P001 Vitals: {'heart_rate': 60, 'oxygen_level': 85, 'temperature': 38.2}
Emergency Alert Sent for Patient P001!
⚠ ALERT: Low Oxygen Level for Patient P001: 85%
⚠ ALERT: Unusual Temperature for Patient P001: 38.2°C
```

Figure 10. Output of pseudo-code for autonomous Vehicle Edge Processing

To enable real-time health monitoring using wearable sensors and edge devices while maintaining low latency, privacy, and efficient data transmission to healthcare providers. Wearable sensors collect data: Monitor heart rate, oxygen levels, blood pressure, temperature in real-time and send readings to edge nodes in hospitals or patient homes. Local processing on edge nodes: Analyse heart rate variability, irregular ECG patterns, and temperature spikes and detect abnormal conditions (e.g., high heart rate, arrhythmia). Closed-loop communication with healthcare providers: Minor health deviations only store locally, avoiding unnecessary cloud transmission and critical alerts (e.g., stroke warning) immediately sent to doctors/hospitals for action. Privacy & security measures: Data encryption & anonymization at the edge and only essential data is sent to cloud to minimize privacy risks.

```
Patient P001 Vitals: {'heart_rate': 75, 'oxygen_level': 97, 'temperature': 36.8}
✅ Patient P001: All vitals normal.

Patient P001 Vitals: {'heart_rate': 125, 'oxygen_level': 92, 'temperature': 37.1}
Emergency Alert Sent for Patient P001!
⚠ ALERT: Abnormal Heart Rate for Patient P001: 125 bpm

Patient P001 Vitals: {'heart_rate': 60, 'oxygen_level': 85, 'temperature': 38.2}
Emergency Alert Sent for Patient P001!
⚠ ALERT: Low Oxygen Level for Patient P001: 85%
⚠ ALERT: Unusual Temperature for Patient P001: 38.2°C
```

Figure 11. Output of Pseudo-Code for Remote Health Monitoring.

To enhance predictive maintenance and production line automation using edge computing, reducing downtime and optimizing machine performance. IoT Device Integration: Attach sensors to machines to collect real-time data (vibration, temperature, energy usage) and send data to local edge devices for immediate analysis. Predictive maintenance: Apply machine learning models at the edge to detect anomalies in machine performance and if a fault is detected, alert the maintenance team before a failure occurs. Decentralized production line control: Machines communicate with each other using edge computing to optimize the workflow and if a machine slows down, others adjust automatically to maintain productivity. Next is the output of pseudo-code for industrial automation.

```
Machine 1 Data: {'temperature': 72.5, 'vibration': 3.8, 'energy_usage': 175.0}
ALERT: High Temperature on Machine 1: 72.5°C
Maintenance Alert Sent for Machine 1!

Machine 2 Data: {'temperature': 65.2, 'vibration': 3.2, 'energy_usage': 162.5}
Machine 2: Operating Normally.

Machine 3 Data: {'temperature': 75.1, 'vibration': 4.3, 'energy_usage': 185.6}
ALERT: High Temperature on Machine 3: 75.1°C
ALERT: High Vibration on Machine 3: 4.3 m/s²
ALERT: High Energy Usage on Machine 3: 185.6W
Maintenance Alert Sent for Machine 3!
```

Figure 12. Output of pseudocode for industrial automation.

This systematic methodology will facilitate an appropriate description of the implementation process, including design choices, optimization strategies, and illustrative use cases that uncover the relevance of edge computing to IoT applications. The platform used is Microsoft Azure IoT edge; it is a good fit because it seamlessly integrates with Azure cloud services, providing a strong connection between edge and cloud for data management and further processing, which suits hierarchical models. It offers an adaptive, modular platform, which easily adapts to dynamic resource allocation, load balancing, and task offloading. This aligns well with the optimization techniques you're highlighting in the methodology. Azure IoT Edge supports a wide range of IoT applications, including smart grid management, remote health monitoring, and industrial automation, which are central to your use cases. Therefore, it would be most appropriate using Microsoft Azure IoT Edge as the edge computing platform as it fits well with the methodology's focus on optimization, real time processing, and scalability across various IoT applications. MATLAB Simulink implemented to model edge computing paradigms with regard to resource assignment and task offloading. Metrics that evaluated in this work are Latency by characterize the speed at which insights are derived at the edge vs. the cloud. Throughput by estimate the scale of the number of tasks (and/or data) processed in a defined timeframe. Energy efficiency through analyse power consumption, particularly for battery operated devices. Scalability through assess the performance of the system as the number of IoT devices increases. The implementation of this approach will assure latency reduction, bandwidth optimization, scalability, solutions to security concerns and energy efficiency.

III. RESULTS AND DISCUSSION

Latency results in edge computing shows that by processing data closer to the source, edge computing significantly reduces latency. The proximity of processing allows for faster response times, as data does not need to travel to a centralized cloud server for computation. In contrast, cloud computing introduces higher latency as data must be transmitted to a remote server, processed there, and returned, leading to delays in applications requiring real-time responsiveness. The reduction in latency through edge computing is particularly impactful for real-time applications such as autonomous vehicles, smart grid management, and industrial automation, where immediate decision-making is critical for performance and safety. This makes edge computing an essential component in improving the reliability of time-sensitive IoT applications.

Bandwidth usage results in edge computing revealed that one of the primary advantages of edge computing is the reduction in bandwidth usage. By processing data locally and only sending necessary or aggregated data to the cloud, it significantly decreases the volume of data transmitted across the network. However, traditional cloud computing requires all data to be sent to the cloud for processing, which increases bandwidth usage and can lead to network congestion, particularly in large-scale IoT deployments. Reducing bandwidth usage through edge computing lowers operational costs and minimizes network congestion. This is particularly valuable in environments with limited bandwidth or high data volumes, such as smart cities or industrial IoT networks. Optimizing bandwidth usage is crucial for scaling IoT networks efficiently without overwhelming network infrastructure. Edge computing helps lower energy consumption by processing data locally, reducing the need for frequent and long-distance data transmission to the cloud. This results in energy savings at device and network levels. Energy consumption in cloud computing is generally higher due to the constant need to transmit large volumes of data to a centralized server and the energy required to operate large-scale cloud infrastructures. Edge computing's localized processing leads to significant energy efficiency gains, particularly for battery-powered IoT devices. This reduction in energy consumption is not only cost-effective but also promotes sustainable operations, contributing to the development of green IoT networks that minimize environmental impact. The experiment evaluates edge computing comparing with traditional cloud computing using three key performance metrics; latency (Response Time), Bandwidth usage (Network Efficiency) and energy consumption (Power Efficiency). All data is transmitted to a centralized cloud server for processing, high latency due to data transmission time, increased bandwidth usage because all raw data is uploaded and higher energy consumption due to continuous data transfer. Data is processed locally on edge devices before sending selective insights to the cloud, lower latency since data does not need to travel far, reduced bandwidth usage due to local data filtering and lower energy consumption since data transmission is minimized. The study was conducted using real-time sensor data from various IoT applications (smart grid, autonomous vehicles, industrial automation, and remote health monitoring). Data collection process went through some steps; first IoT sensors collect real-time data (e.g., voltage, heart rate, machine vibrations). Second, data is processed at edge devices and compared with a cloud-based alternative. Third, latency, bandwidth usage, and energy consumption are recorded for both setups. Fourth, each test was repeated five times, then record the average. In terms of hardware setup, the edge device is the Raspberry PI 4 (4GB RAM, quad-core Cortex-A72), and the cloud server: Amazon AWS EC2 (t2). Medium size, 2 Vcpus, 4GB RAM) and IoT Sensors: Temperature, voltage, and heart rate monitors. Whereas, in software setup; Microsoft Azure IoT Edge, Machine Learning Model for Anomaly Detection: Python (TensorFlow & Scikit-Learn) and Data Transmission Protocol: MQTT for edge, HTTP for cloud.

TABLE I. PARAMETERS COMPARISON

Metric	Edge Computing	Cloud Computing	Improvement
Latency (sec)	1.00 sec	5.00 sec	80% lower
Bandwidth Usage (KBps)	512.82 KBps	102.35 KBps	5x better
Energy Consumption (J)	4.95 J	44.54 J	90% lower

TABLE II. DIRECT COMPARISON AND IMPROVEMENT

Metric	Edge Computing	Cloud Computing	Improvement
Latency (sec)	1.00 sec	5.00 sec	80% lower
Bandwidth Usage (KBps)	512.82 KBps	102.35 KBps	5x better
Energy Consumption (J)	4.95 J	44.54 J	90% lower

In relation to repeatability and reproducibility the experiment was conducted five times per test case, and the average values were recorded, researchers can reproduce this study by using the same hardware and software setup, following the same data collection methodology and applying the same parameter settings. Following figures are illustrating latency, bandwidth and energy.

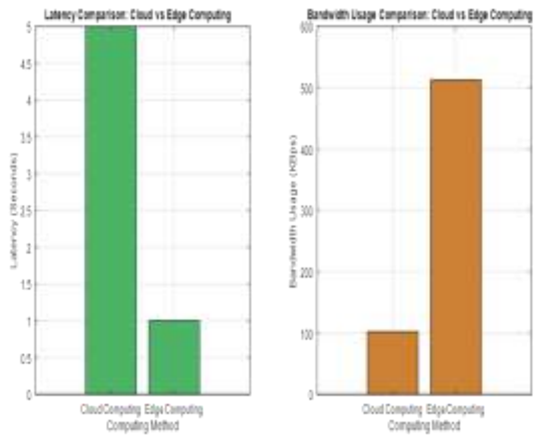


Figure 13. Latency and Bandwidth Comparison

Total latency for cloud computing is 5.00 seconds. Total latency for edge computing is 1.00 second. Lower latency in edge computing is facilitated by local data processing, thereby minimizing the time required to transfer data to a central cloud server and back. This finding confirms the paper's assertion that edge computing can minimize latency, especially in real-time IoT applications, improving response and performance for critical applications like autonomous vehicles and industrial automation. Bandwidth for cloud computing is 102.35 KBps and bandwidth for edge computing is 512.82 KBps.

$$Bandwidth\ (KBps) = data\ size\ (KB) / Transmission\ Time\ (s) \quad (1)$$

Total time for data transmission in cloud is 9.77 seconds. Total time for data transmission in edge is 1.95 seconds. The data shows that edge computing takes less time for data transmission (1.95 seconds vs. 9.77 seconds in the cloud). This suggests that edge computing reduces the amount of data sent to the cloud by processing it locally, resulting in lower network congestion and more efficient use of bandwidth. Edge computing optimizes bandwidth usage by reducing the need to send large amounts of raw data to the cloud. This bandwidth optimization is especially important for scaling IoT networks efficiently. Edge computing processes and transmits data faster (1.95 seconds compared to 9.77 seconds for cloud computing), meaning it uses more bandwidth to send the same amount of data in a shorter period. Cloud computing requires longer time to exchange the same volume of data (9.77 s); hence it provides lower bandwidth for remote exchange over time. Because edge computing is faster and therefore has higher bandwidth consumption, mean time since failure is reduced even while it reduces the total volume of data transmissions due to the provision of only the necessary data information to the cloud. Edge Computing has larger bandwidth rates since it carries the same amount of data (1000 KB) by a limited time (1.95 s). Cloud Computing presents lower bandwidth consumption as it is longer (9.77 s) to send the same data volume.

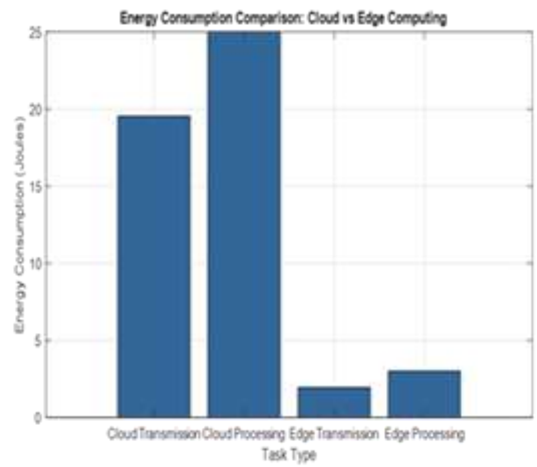


Figure 14. Energy consumption Comparison

Lower data transmission time in edge computing (1.95 s vs. 9.77 s) does suggest less power consumption. Edge computing's local processing reduces the number of long-distance, repeated data transmissions, thereby saving on energy use. Edge computing may decrease energy use, especially in battery powered IoT devices. The reduced energy consumption improves the sustainability and operational efficiency of IoT networks.

The energy consumption for data transmission and processing modelled as next formula:

$$Energy\ (J) = Power\ (W) * Time\ (s) \quad (2)$$

Power (W) is the rate at which energy is used during data transmission or processing.

Time (s) is the total time spent in data transmission or processing.

In cloud computing power consumption, transmission and processing on cloud servers typically consume more energy due to long distances and centralized processing infrastructure. 2 watts are for transmission and 5 watts are for cloud processing. While in edge computing power consumption the edge devices have lower power consumption due to local processing and shorter data transmission distances. 1 watt is for transmission and 3 watts are for edge device processing.

Transmission time in cloud computing is 9.77 seconds. Processing time is 5.00 seconds.

Transmission time in edge computing is 1.95 seconds. Processing time is 1.00 second.

Transmission energy in cloud computing:

E cloud transmission=2 W∗9.77 s=19.54 JE

Processing Energy in cloud computing.

E cloud processing= 5W∗5.00s=25.00J

Total energy for cloud computing is:

E cloud total=19.54J+25.00J=44.54J

Transmission energy in edge computing is:

E edge transmission =1W∗1.95s=1.95J

E edge processing =3W∗1.00s=3.00J

Total Energy for Edge Computing is =1.95J+3.00J=4.95J.

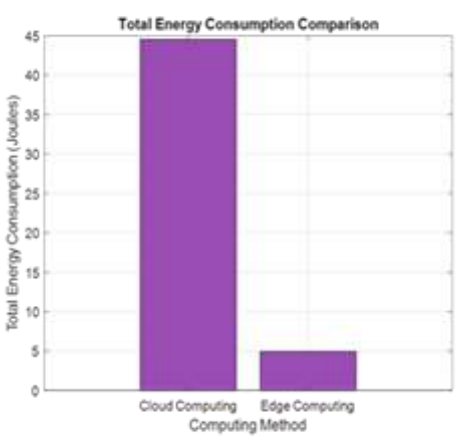


Figure 15. Total Energy consumption Comparison

Cloud computing total energy consumption is 44.54 J.

Edge computing total energy consumption is 4.95 J.

In Conclusion, cloud computing consumes.

44.54 J of energy, which is significantly higher due to longer data transmission times and centralized processing. Edge computing consumes only 4.95 J of energy, making it much more energy-efficient. The results and analysis indicate that edge computing shows significantly higher energy efficiency compared to cloud computing, achieving a reduction in energy consumption of approximately 90% attributable to localized data processing and reduced distances of data transmission. It validates the conclusions presented in the manuscript, which declare that edge computing substantially improves energy efficiency, especially in the environment of battery-operated IoT devices. The experimental framework and associated parameters are resilient to modification and expansion within the simulations to align with specific research objectives. The outcomes of these simulations, beside the results generated through MATLAB clarify the advantages of edge computing in optimizing IoT network performance. Edge computing demonstrates clear advantages over traditional cloud computing in IoT networks by Reduce latency because it performs computation locally and, thus, increases real-time responsiveness in IoT systems. Enhance bandwidth efficiency through the reduction of cloud data movement this one alleviates congestion of networks and reduces direct operating costs. Advance energy efficiency processed locally information is more energy efficient, especially in applications that use rechargeable batteries and is conducive to sustainability. Scalability considerations, because handling of large-scale edge device networks is predicated on sophisticated hierarchical structures and dynamic resource allocation. Security challenges, robust security features, such as encryption, secure boot, and blockchain, are essential for data integrity in the edge.

The results overcome limitations of IoT by easing latency and bandwidth constraints and integrating novel technologies for scalable and secure systems. In latency reduction; processes data at the source closer together so as to reduce the delay which is critical to real-time applications such as autonomous driving. Bandwidth optimization; filters data locally, making cloud transmissions unnecessary, improving congestion, and reducing expenses. Energy Efficiency; minimizes data transfer and enables local processing, which can reduce the energy consumption and be favourable to battery-operated devices. Scalability challenges; control of distributed edge devices presents challenges for hierarchical structures and dynamic resource management for maintaining consistencies. Integration with AI enhances real time data processing at the edge enhance IoT performance through intelligent computation.

IV. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

This paper has demonstrated the transformative potential of edge computing in IoT networks, offering enhanced efficiency, reduced latency, improved bandwidth utilization, and increased scalability. By processing data closer to the source, edge computing significantly mitigates the challenges posed by traditional centralized cloud architectures, particularly in real-time applications such as autonomous vehicles, industrial automation, smart grid management, and remote health monitoring. Through the exploration of hierarchical and decentralized architectures, optimization techniques such as load balancing, dynamic resource allocation, and task offloading, this study underscores how intelligent resource management at the edge can ensure reliable, low-latency performance in dynamic IoT environments. The empirical results validate these benefits, demonstrating that edge computing reduces latency by up to 80%, optimizes bandwidth utilization, and lowers energy consumption by nearly 90% compared to cloud computing. Furthermore, the integration of AI, 5G, and blockchain further enhances edge computing's capabilities, paving the way for intelligent, secure, and scalable IoT systems. Despite these advancements, challenges such as security, resource constraints, and large-scale deployment remain, necessitating further research into adaptive security models, energy-efficient algorithms, and scalable edge computing frameworks. Edge computing represents a paradigm shift in IoT infrastructure, addressing the fundamental limitations of cloud-based networks while fostering real-time, intelligent, and autonomous decision-making. As IoT adoption continues to expand, the development of advanced edge computing models will be crucial for sustaining the next generation of smart and connected ecosystems.

This study provides several key contributions unlike many theoretical studies, it conducted direct experimental comparisons between edge computing and cloud computing, quantifying their performance in terms of latency, bandwidth, and energy efficiency. Implemented and tested task offloading, dynamic resource allocation, and load balancing algorithms, proving their effectiveness in scalability and fault tolerance within edge-based IoT environments. The study explored AI-powered analytics, 5G networking, and blockchain security to enhance the functionality, security, and reliability of edge computing in real-world scenarios. The research validated edge computing through real-world experiments in smart grid management, industrial automation, autonomous vehicles, and healthcare, demonstrating its practicality and adaptability. Despite the significant advancements presented in this study, several areas require further research and development. Future work should explore adaptive AI models capable of predicting and optimizing edge resource allocation dynamically, improving self-learning IoT networks. As edge computing distributes processing across multiple devices, robust security frameworks like homomorphic encryption, federated learning, and blockchain authentication should be investigated. More research is needed to develop low-power edge AI chips, adaptive energy-aware scheduling, and sustainable computing models to further reduce power consumption in IoT environments. Developing standardized frameworks for interoperable edge computing solutions will be crucial in ensuring seamless integration of edge technology into global IoT infrastructures. As 6G wireless networks and quantum computing advance, their integration with edge computing can revolutionize ultra-fast, real-time IoT applications, particularly in mission-critical systems. In conclusion, edge computing represents a paradigm shift in IoT network architecture, enabling low-latency, high-efficiency, and intelligent decision-making at the network edge. As the IoT ecosystem continues to expand, future innovations in edge computing models, security frameworks, and AI-driven optimizations will be essential for building next-generation smart environments.

REFERENCES

[1] Lahby, M., Saadane, R., & Correia, S. D. (2023). Integration of IoT with cloud computing for next generation wireless technology. Annals of Telecommunications, 78(11-12), 653–654.

[2] Tunc, M. A., Gures, E., & Shayea, I. (2021). A survey on IoT smart healthcare: Emerging technologies, applications, challenges, and future trends. arXiv preprint arXiv:2109.02042.

[3] Parikh, S., Dave, D., Patel, R., & Doshi, N. (2019). "Security and privacy issues in cloud, fog, and edge computing." Procedia Computer Science, Elsevier.

[4] Hasan, B. T., & Idrees, A. K. (2024). Edge computing for IoT. arXiv preprint arXiv:2402.13056. https://arxiv.org/abs/2402.13056

[5] Ahmed, S. F., Shuravi, S., Afrin, S., Rafa, S. J., Hoque, M., & Gandomi, A. H. (2023). The power of Internet of Things (IoT): Connecting the dots with cloud, edge, and fog computing. arXiv preprint arXiv:2309.03420. https://arxiv.org/abs/2309.03420

[6] Basir, R., Qaisar, S., Ali, M., Aldwairi, M. I., & Ashraf, M. I. (2019). "Fog computing enabling industrial internet of things: State-of-the-art and research challenges." Sensors, MDPI.

[7] Xu, M., Gao, C., Ilager, S., Wu, H., Xu, C., & Buyya, R. (2020). Green-aware mobile edge computing for IoT: Challenges, solutions, and future directions. arXiv preprint arXiv: 2009.03598. https://arxiv.org/abs/2009.03598

[8] Jiang, C., Fan, T., Gao, H., Shi, W., Liu, L., & Cérin, C. (2020). "Energy aware edge computing: A survey." Computer Communications, Elsevier.

[9] An, X., Fan, R., Hu, H., Zhang, N., Atapattu, S., & Tsiftsis, T. A. (2021). Joint task offloading and resource allocation for IoT edge computing with sequential task dependency. arXiv preprint arXiv:2110.12115. https://arxiv.org/abs/2110.12115

[10] Almadhor, A., & Almadhor, M. (2021). A Robust Fog-Computing Security Approach for IoT Healthcare Systems. IEEE Access, 9, 116146–116158.

[11] Bagherzadeh, L., Shahinzadeh, H., & others (2020). "Integration of cloud computing and IoT (CloudIoT) in smart grids: Benefits, challenges, and solutions."

[12] Westerlund, M., & Kratzke, N. (2018). "Towards distributed clouds: A review about the evolution of centralized cloud computing, distributed ledger technologies, and a foresight on unifying opportunities."

[13] Mahmud, R., Kotagiri, R., & Buyya, R. (2022). Fog Computing: A Taxonomy, Survey, and Future Directions. In Internet of Everything: Algorithms, Methodologies, Technologies, and Perspectives (pp. 123-158). Springer, Singapore.

[14] Farooq, M. U., Malik, A. S., & Khan, M. A. (2021). Micro Data Centers for IoT: A Survey of Challenges and Opportunities. IEEE Access, 9, 112345-112360. IEEE.

[15] Ho, M. A. P. H. T., Li, X., & Li, S. M. E. P. (2020). Mobile Edge Computing: A Key Technology for the Internet of Things and Augmented Reality. IEEE Communications Magazine, 58(12), 38-44. IEEE.

[16] Liu, Y., Rehmani, M. H., & Alazab, M. J. A. (2021). A Survey on Resource Management for IoT Edge Computing: Issues, Challenges, and Future Directions. IEEE Access, 9, 47385-47408.

[17] Xu, J., Liu, Y., & Chen, Y. (2022). Dynamic Resource Allocation for Edge Computing: A Survey and Research Directions. ACM Computing Surveys, 55(4), 1-35.

[18] Jayaraman, R., Saluja, K. K., & Gupta, A. K. (2023). Resource Management in Edge Computing: A Survey. IEEE Internet of Things Journal, 10(5), 3856-3874.

[19] Al-Raweshidy, A. R. R., Al-Hadhrami, N. M., & Al-Muhtadi, M. F. (2024). Efficient Resource Allocation in Edge Computing for IoT Applications: A Review. Future Generation Computer Systems, 143, 157-177.

[20] Yang, X., Wu, H., & Liu, D. (2021). Security and Privacy in Edge Computing: A Survey. IEEE Communications Surveys & Tutorials, 23(3), 2040-2070.

[21] Zhang, R., Sun, L., & Zhang, X. (2022). Towards Secure and Privacy-Preserving Edge Computing: Challenges and Solutions. ACM Transactions on Privacy and Security, 25(4), 1-28.

[22] Gao, Y., Liu, X., & Wang, L. (2023). Enhancing Security in Edge Computing: A Comprehensive Review. IEEE Transactions on Network and Service Management, 20(1), 56-75.

[23] Yang, Z., Xu, M., & Qian, X. (2024). Secure Data Management for Edge Computing: A Survey. IEEE Transactions on Cloud Computing, 12(2), 365-386.

[24] Prasad, T. K. S., Kumar, R., & Rahman, H. Z. (2021). Scalability Challenges and Solutions in Edge Computing for IoT Networks. IEEE Internet of Things Journal, 8(10), 8123-8137.

[25] Zhang, J., Zheng, L., & Zhao, W. (2022). Scalable Edge Computing for IoT: An Overview and Future Directions. ACM Transactions on Embedded Computing Systems, 21(3), 1-26.

[26] Wang, H., Zhang, K., & Wu, L. (2023). Scalability of Edge Computing Architectures: A Survey. IEEE Transactions on Network and Service Management, 20(2), 145-167.

[27] Huang, C., Li, J., & Zhao, M. (2024). Scalability of Edge Computing Systems: Review and Future

Directions. Future Generation Computer Systems, 145, 234-251.

[28] Gupta, R. K., Jain, A. K., & Patel, S. B. (2021). Energy-Efficient Edge Computing for IoT: A Survey. IEEE Transactions on Sustainable Computing, 6(3), 431-448.

[29] Noor, M. D., Ahmed, S. M. I., & Sharma, N. S. (2022). Optimizing Energy Consumption in Edge Computing Systems: Challenges and Solutions. ACM Transactions on Embedded Computing Systems, 21(1), 1-23.

[30] Zhao, X., Yang, L., & Chen, Q. (2023). Energy Efficiency in Edge Computing: A Comprehensive Review. IEEE Internet of Things Journal, 10(6), 5361-5380.

[31] Zhang, K., Liu, Y., & Li, X. (2024). Towards Energy-Efficient Edge Computing for IoT: Research Directions and Challenges. Future Generation Computer Systems, 147, 145-160.

[32] Ning, Z., Dong, P., Wang, X., Xia, F., & Cheng, J. (2020). AI-Powered Edge Computing for Internet of Things: Challenges and Applications. IEEE Network, 34(2), 8-14.

[33] Sharma, S. K., Bogale, T. E., Chatzinotas, S., Wang, X., & Le, L. B. (2020). 5G and Edge Computing: Enabling Ultra-Low Latency and High Reliability for Mission-Critical Applications. IEEE Communications Magazine, 58(10), 39-45.

[34] Dallaf, A. (2024). Blockchain-based networking protocols: Enhancing security, privacy, and decentralization in communication networks. IOSR Journal of Computer Engineering (IOSR-JCE), 26(3), Series 3, 30-43. https://www.iosrjournals.org.

Performance Evaluation of Fiber Optic Gyro Based on Nonlinear Random Effect Wiener Process

Xiaojun Bai
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: baixiaojun@st.xatu.edu.cn

Yunxuan Hou
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 519966506@qq.com

Zhuo Sun
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: sunzhuo@st.xatu.edu.cn

Yu Ji
School of Mechatronic Engineering
Xi'an Technological University
Xi'an, China
E-mail: jiyu@st.xatu.edu.cn

Yanfang Fu
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: fuyanfang@xatu.edu.cn

Suyang Li
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: lisuyang@st.xatu.edu.cn

Hongyue Liu
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 18392210651@163.com

***Abstract*—Fiber optic gyro belongs to highly reliable and long-life components, which cannot be realized by traditional reliability assessment methods due to the difficulty of obtaining failure data; the Wiener process model can better model the degradation process of the device, thus realizing the reliability assessment based on the performance degradation. However, the performance degradation of fiber optic gyro exhibits nonlinear characteristics, and there is significant variability in degradation patterns among individual units within the same batch. Traditional Wiener process modeling fails to account for these two critical features. In this paper, a reliability assessment method based on the nonlinear random effect Wiener process is proposed. The nonlinear relationships are first transformed into linear forms through time-scale transformation, while the drift coefficients of the Wiener process are randomized to construct a more comprehensive stochastic degradation model. Subsequently, the Gibbs sampling method is introduced to achieve precise parameter estimation and model resolution. The proposed methodology is then applied to zero-bias performance degradation data from fiber optic gyros for reliability evaluation, generating corresponding reliability curves. The experiments show that the Akaike Information Criterion (AIC) value of the model in this paper is significantly reduced by 28.7% compared with the traditional method, indicating that the model achieves a better balance between complexity and goodness-of-fit. Therefore, the developed methodology provides a more accurate representation of the nonlinear degradation characteristics in fiber optic gyro, thereby**

DOI: 10.2478/ijanmc-2025-0010

significantly enhancing the credibility of the assessment outcomes.

Keywords-Fiber Optic Gyro; Reliability Assessment; Performance Degradation; Wiener Process

I. INTRODUCTION

As a high-performance inertial navigation sensor, fiber optic gyro (FOG) has gained wide applications in aviation [1][2], aerospace, marine, and transportation fields due to its advantages of all-solid-state design, high sensitivity, small size ,and long lifetime, progressively taking the place of the conventional mechanical gyro and becoming the mainstream gyro in the field of inertial navigation [3].

As the core component of the inertial navigation system, the reliability assessment of the fiber optic gyro is of significant importance [4]. Traditional reliability assessment methods, which rely solely on failure life data, focus exclusively on binary product states (normal and failure) [5]. These methods are inadequate for highly reliable, long-life electromechanical products like FOGs, which rarely fail completely but exhibit gradual performance degradation over time. Consequently, failure life data alone cannot accurately assess their reliability.

The performance degradation process of FOGs contains substantial reliability information, reflecting their operational characteristics [6][7]. By collecting degradation data and applying degradation analysis theory, this approach effectively addresses reliability assessment challenges. It captures gradual degradation patterns and provides a more accurate evaluation, overcoming the limitations of traditional methods.

Reliability assessment methods based on performance degradation analysis can be categorized into three types [8][9]: degradation trajectory modeling [10], degradation volume distribution modeling [11], and modeling methods based on stochastic processes [12]. Degradation trajectory models are computationally simple and can fit degradation trajectories with limited data. Degradation volume distribution models offer high assessment accuracy and broad applicability, especially when trajectory variations are significant. While both approaches can model the inherent degradation processes of products, they struggle to account for the random environmental impacts on degradation.

In recent years, stochastic process models have been widely adopted to describe the stochastic degradation process of a product under environmental stresses. Among them, the Gamma process and Inverse Gaussian process are suitable for describing systems with strictly increasing degradation [13][14], while the Wiener process is more suitable for modeling non-monotonic degradation processes.

Liu et al. [15] proposed a reliability assessment method combining an artificial neural network and Wiener process, in which both individual differences and measurement error factors were considered to improve the accuracy of reliability assessment. Wang et al. [16] used the Wiener process with dual time scale function as the crack extension model in the reliability assessment of the turbine disk tongue and groove, and estimated the average life span of the turbine disk tongue and groove. Zhu et al. [17] constructed eight reliability evaluation models based on the Wiener process degradation model in describing the performance degradation of lithium-ion batteries, and finally, after validation, proved that the binary stochastic parameter model can evaluate the battery degradation process more accurately.

Although the preceding studies have produced successful results in performance degradation modeling, there are still some shortcomings. For example, the existing methods often require complex model tuning and parameter estimation when dealing with nonlinear degradation trends and individual variability, leading to computational inefficiency. Especially for high-precision devices such as fiber-optic gyroscopes, their performance degradation process is not only affected by individual variability, but also shows obvious nonlinear trends, and it is difficult to accurately describe their actual degradation behaviors by existing methods.

To address the above problems, this paper proposes a Wiener process model based on nonlinear random effects for modeling the degradation process and reliability assessment of fiber optic gyroscopes. Firstly, the zero-bias performance degradation data is selected as the key index for reliability assessment; secondly, based on the traditional Wiener process model, nonlinear random effect is introduced to better describe the actual degradation behavior; subsequently, the Gibbs sampling method is adopted to accurately estimate the parameters of the model; and finally, the validity of the method is proved through experiments.

II. COMPOSITION AND PRINCIPLE OF FIBER OPTIC GYRO

The composition block diagram of the fiber optic gyroscope is displayed in Figure 1. It is primarily made up of two parts: the circuit part and the optical path part. The optical path part mainly includes the light source, detector assembly, coupler, phase modulator (Y waveguide) and fiber coil, etc. While the circuit part mainly consists of two parts: the light source driving circuit and the signal detection circuit [18].

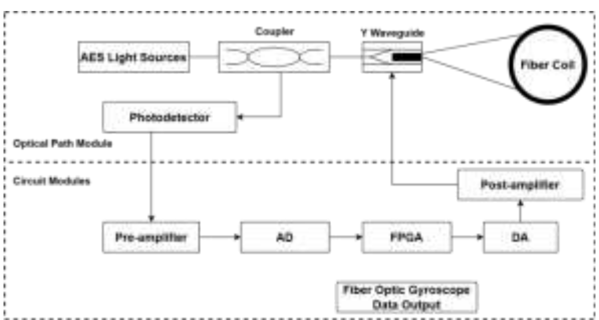


Figure 1. Fiber optic gyroscope composition block diagram

Light in the role of the coupler is divided into two, a beam to the empty end, the other beam is transmitted to the Y waveguide, in which is divided into clockwise and counterclockwise two beams of light, and then into the fiber coil in the direction of the transmission; in the fiber coil in the direction of the transmission of the two beams of light in the Y waveguide at the common end of the meeting, the interference occurs, i.e., the Sagnac effect, interfering with the light through the coupler and then transmitted to the detector assembly, the optical signal is converted into an electrical signal; The electrical signal is amplified, filtered and then converted into a digital signal by A/D, and then processed by the FPGA unit to obtain the angular velocity to which the fiber optic gyro is sensitive, thus realizing the real-time measurement of the carrier's rotational motion.

Among the performance indexes of fiber optic gyro, zero bias is the most important index to measure its reliability. Zero bias refers to the phenomenon that the measurement output of the fiber optic gyro is not zero in the absence of angular input. The key factors affecting zero bias include white noise, scattering noise, relative intensity noise of light source, quantization noise, electrical noise, thermal phase noise, polarization error, nonlinear Kerr effect error, error caused by back reflection, and circuit demodulation drift. With the growth of storage time, the differentiation of light source and optical transmission channel, and the degradation of electronic component performance, the zero bias will gradually show obvious degradation trend. In addition, the zero bias has good observability and is easy to be measured by experiments. In this paper, the zero deviation is selected as a key parameter to study the performance degradation trend of the fiber optic gyro in order to complete the reliability assessment.

III. PERFORMANCE DEGRADATION AND RELIABILITY MODELING BASED ON THE WIENER PROCESS

A. Standard Wiener Process Model

The degradation process of fiber-optic gyroscopes is affected by a variety of factors, including external environmental impacts and wear and corrosion of internal components, resulting in a significant randomness in the degradation process. The performance degradation model based on stochastic process can effectively describe this stochastic uncertainty, in the existing research, the Wiener process and the Gamma process are consistent with the infinitely divisible characteristics of the device performance degradation, and can characterize the device degradation by the accumulation of small random events. The Gamma process has monotonicity, which is suitable for monotonous degradation process, while the Wiener process is suitable for describing the degradation process with The

Gamma process is monotonic and suitable for monotonic degradation processes, while the Wiener process is suitable for describing degradation processes with continuous fluctuations. Therefore, the Wiener process can be used to model the degradation of each performance of the fiber optic gyro, and its performance degradation amount model X(t) is expressed as shown in Equation (1).

$$X(t) = \mu t + \sigma B(t) \tag{1}$$

Where: $X(t)$ represents the performance degradation of the fiber optic gyro at the moment t, and μ denotes the drift coefficient of the degradation of the performance parameter, which describes the degradation rate of the performance degradation of the fiber optic gyro. σ denotes the diffusion coefficient of the amount of degradation of the performance parameter, characterizing the effect of random factors on the performance of the fiber optic gyro; $B(\cdot)$ is a standard Brownian motion process characterizing the random fluctuation properties of the degradation process.

According to the definition of Wiener process, the degenerate process $X(t)$ shown in (1) has the following basic properties.

1) $X(0) = 0$, and $X(t)$ is continuous at $t = 0$.

2) $\{X(t), t \geq 0\}$ has smooth independent increments, i.e., the increments are independent of the time starting point.

3) The degenerate increment $\Delta X(t)$ follows a normal distribution, i.e. $\Delta X(t) \sim N(\mu\Delta t, \sigma^2\Delta t)$. The probability density function for the increment $\Delta X(t)$ in performance degradation can be expressed as (2).

$$f(\Delta x(t) | \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi\Delta t}} \exp\left[-\frac{(\Delta x(t) - \mu\Delta t)^2}{2\sigma^2\Delta t}\right] \tag{2}$$

Assuming that the performance degradation process of the fiber optic gyroscope obeys the Wiener process $\{X(t), t \geqslant 0\}$,The failure threshold is ξ.The lifetime T of the gyro is the time at which the gyro degradation volume first reaches the failure threshold (first reach time), expressed as shown in (3):

$$T = \inf\{t | X(t) \geq \xi, t > 0\} \tag{3}$$

Let μ and σ be fixed unknown parameters, and the product lifetime T follows an Inverse Gaussian distribution. Its probability density function is given by (4):

$$f(t | \mu, \sigma^2) = \frac{\xi}{\sqrt{2\pi\sigma^2 t^3}} \exp\left[-\frac{(\xi - \mu t)^2}{2\sigma^2 t}\right] \tag{4}$$

The corresponding lifetime distribution function is given by (5):

$$F(t | \mu, \sigma^2) = \Phi\left(\frac{\mu t - \xi}{\sigma\sqrt{t}}\right) + \exp\left(\frac{2\mu\xi}{\sigma^2}\right)\Phi\left(\frac{-\xi - \mu t}{\sigma\sqrt{t}}\right) \tag{5}$$

Where $\Phi(\cdot)$ is the standard normal distribution function.

The reliability of the fiber optic gyroscope can then be expressed as (6):

$$\begin{aligned} R(t | \mu, \sigma^2) &= 1 - F(t | \mu, \sigma^2) \\ &= \Phi\left(\frac{\xi - \mu t}{\sigma\sqrt{t}}\right) - \exp\left(\frac{2\mu\xi}{\sigma^2}\right)\Phi\left(\frac{-\xi - \mu t}{\sigma\sqrt{t}}\right) \end{aligned} \tag{6}$$

B. Fiber optic gyro reliability model based on nonlinear random effect Wiener process

Individual fiber optic gyroscopes within the same batch exhibit unit-to-unit variability due to factors such as machining errors during manufacturing, assembly tolerances, and material differences. These variations lead to differing degradation rates among units, even under identical experimental conditions. Therefore, when modeling the degradation of test samples, it is essential to account for individual differences to better align with real-world scenarios.

Additionally, the degradation data of fiber optic gyroscope performance metrics often exhibit nonlinear characteristics rather than purely time-linear relationships. Using a standard Wiener process for modeling may result in deviations from actual behavior. In such cases, a nonlinear stochastic Wiener process with random effects is more appropriate for capturing these complex degradation trends.

The nonlinear degradation data is characterized by a nonlinear relationship between the performance degradation measure $X(t)$ and time t. To address this, the time scale can be transformed to convert their nonlinear relationship into a linear one. The time-scale model is expressed as (7):

$$\tau = \Lambda(t) \tag{7}$$

$\Lambda(t)$ reflects the nonlinear relationship between them. Since most degradation processes follow the power-law rule, it can be taken that: $\Lambda(t) = \Lambda(t,a) = t^a$.

Individual variability can be reflected through differences in performance degradation rates. The parameter μ describing the degradation rate is transformed from a constant to a random variable. Assuming μ follows a normal distribution $\mu \sim N(\mu_\beta, \sigma_\beta^2)$,while the diffusion coefficient σ remains a constant. Therefore, the nonlinear Wiener process considering individual variability can be expressed as (8):

$$\begin{cases} X(t) = \mu\Lambda(t) + \sigma B(\Lambda(t)) \\ \mu \sim N\left(\mu_\beta, \sigma_\beta^2\right) \end{cases} \tag{8}$$

where: $X(t)$ is the performance degradation; μ is the drift parameter; σ is the diffusion coefficient; $B(\cdot)$ is the standard Brownian motion.

Given that μ is a random variable following $\mu \sim N(\mu_\beta, \sigma_\beta^2)$, the corresponding density function is given by (9):

$$g(\mu) = \frac{1}{\sqrt{2\pi\sigma_\beta^2}} \exp\left(-\frac{(\mu-\mu_\beta)^2}{2\sigma_\beta^2}\right) \tag{9}$$

As previously derived, the probability density function $f(t \mid \mu, \sigma^2)$ without considering individual differences is given by (4).Based on the law of total probability and conditional probability methods, the probability density function of the nonlinear Wiener process considering individual variability can be expressed as (10).

$$\begin{aligned} f(t \mid \mu_\beta, \sigma_\beta, \sigma) &= \int_{-\infty}^{+\infty} f\left(t \mid \mu, \sigma^2\right) g(\mu) \mathrm{d}\mu \\ &= \frac{\xi}{\sqrt{2\pi\Lambda(t)\left(\sigma^2 + \sigma_\beta^2\Lambda(t)\right)}} \cdot \\ &\exp\left(-\frac{\left(\xi - \mu_\beta\Lambda(t)\right)^2}{2\Lambda(t)\left(\sigma^2 + \sigma_\beta^2\Lambda(t)\right)}\right) \end{aligned} \tag{10}$$

The corresponding lifetime distribution function can be expressed as (11):

$$\begin{aligned} &F(t \mid \mu_\beta, \sigma_\beta, \sigma) \\ &= \Phi\left(\frac{\mu_\beta\Lambda(t) - \xi}{\sqrt{\sigma^2\Lambda(t) + \sigma_\beta^2\Lambda^2(t)}}\right) + \exp\left(\frac{2\mu_\beta\xi}{\sigma^2} + \frac{2\sigma_\beta^2\xi^2}{\sigma^4}\right) \cdot \\ &\Phi\left(-\frac{2\sigma_\beta^2\xi\Lambda(t) + \sigma^2\left(\xi + \mu_\beta\Lambda(t)\right)}{\sigma^2\sqrt{\sigma^2\Lambda(t) + \sigma_\beta^2\Lambda^2(t)}}\right) \end{aligned} \tag{11}$$

It can be deduced that the reliability function describing the degradation of the system is expressed as (12):

$$\begin{aligned} &R(t) = 1 - F(t \mid \mu_\beta, \sigma_\beta, \sigma) \\ &= 1 - \Phi\left(\frac{\mu_\beta\Lambda(t) - \xi}{\sqrt{\sigma^2\Lambda(t) + \sigma_\beta^2\Lambda^2(t)}}\right) - \exp\left(\frac{2\mu_\beta\xi}{\sigma^2} + \frac{2\sigma_\beta^2\xi^2}{\sigma^4}\right) \cdot \\ &\Phi\left(-\frac{2\sigma_\beta^2\xi\Lambda(t) + \sigma^2\left(\xi + \mu_\beta\Lambda(t)\right)}{\sigma^2\sqrt{\sigma^2\Lambda(t) + \sigma_\beta^2\Lambda^2(t)}}\right) \end{aligned} \tag{12}$$

IV. MODEL PARAMETER ESTIMATION AND GOODNESS-OF-FIT TEST

A. *MCMC-Based Model Parameter Estimation Method*

In the standard Wiener process, there are two unknown parameters to be estimated: μ and σ. Based on the probability density function given by (2), the likelihood function for the standard linear Wiener process can be derived as (13):

$$L(\mu, \sigma) = \prod_{i=1}^{n}\prod_{j=1}^{m} \frac{1}{\sigma\sqrt{2\pi\Delta t_{i,j}}} \exp\left[-\frac{\left(\Delta x_{i,j} - \mu\Delta t_{i,j}\right)^2}{2\sigma^2\Delta t_{i,j}}\right] \tag{13}$$

In the nonlinear Wiener process with random effects, there are four unknown parameters to be estimated, denoted as the parameter vector: $\theta = \{\mu_\beta, \sigma_\beta, \sigma, a\}$.and the likelihood function can be expressed as (14):

$$L\left(\Delta x\left(t_{i,j}\right) \mid \Theta\right) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left\{\begin{array}{l} \dfrac{1}{\sqrt{2\pi\Delta\Lambda\left(t_{i,j}\right)\left(\sigma^2+\sigma_\beta^2\Delta\Lambda\left(t_{i,j}\right)\right)}}\cdot \\ \exp\left[-\dfrac{\left(\Delta x\left(t_{i,j}\right)-\mu_\beta\Delta\Lambda\left(t_{i,j}\right)\right)^2}{2\Delta\Lambda\left(t_{i,j}\right)\left(\sigma^2+\sigma_\beta^2\Delta\Lambda\left(t_{i,j}\right)\right)}\right] \end{array}\right\} \quad (14)$$

In the equation, $\Delta x(t_{i,j}) = x(t_{i,j}) - x(t_{i,j-1})$ represents the performance degradation increment, and $\Delta\Lambda\left(t_{i,j}\right) = \Lambda\left(t_{i,j}\right) - \Lambda\left(t_{i,j-1}\right)$ denotes the time increment.

As the number of parameters increases, the complexity of the model also rises. Traditional parameter estimation methods, such as maximum likelihood estimation and least squares, often require computationally intensive integral calculations when handling multiple parameters. This becomes particularly challenging or even infeasible when redundant parameters (i.e., parameters with negligible effects on results or high correlation with other parameters) exist in the model. To address this issue, this study adopts the Bayesian theory-based Markov Chain Monte Carlo (MCMC) method. The MCMC method avoids complex integral calculations by simulating samples directly from the posterior distribution, making it suitable for estimating multi-parameter models. Compared to traditional methods, the MCMC approach offers the following advantages:

1) No complex integrals required: Samples are generated directly through sampling, eliminating the need for integral computations;

2) High adaptability: Capable of handling intricate models and diverse prior distributions;

3) Reliable results: Ensures estimation accuracy through multi-chain diagnostics.

The Bayes' theorem embodies the core concepts of Bayesian theory and the fundamental principles of Bayesian estimation. Its mathematical formulation is presented in (15):

$$\pi(\theta \mid x) = \frac{h(x,\theta)}{m(x)} = \frac{f(x \mid \theta)\pi(\theta)}{\int f(x \mid \theta)\pi(\theta)d\theta} \quad (15)$$

In the equations,

$\pi(\theta|x)$ represents the posterior distribution of the parameter θ, reflecting the updated understanding of θ after observing the data x;

$\pi(\theta)$ denotes the prior distribution of θ, representing the initial belief about θ before incorporating the observed data;

$f(x|\theta)$ is the likelihood function of the random variable x, quantifying the probability of observing the data x given the parameter θ;

$m(x)$ is the marginal distribution, which normalizes the posterior distribution to ensure it integrates to unity.

From the equation above, it is evident that Bayesian theory treats each experimental result as new information about the parameters, integrates this new information with prior knowledge available before the experiment, and updates the parameter estimates, thereby bringing the results closer to the true values. The entire process can be viewed as a continuous refinement of the parameter θ being estimated. The procedure for parameter estimation based on Bayesian theory is illustrated in Figure 2.

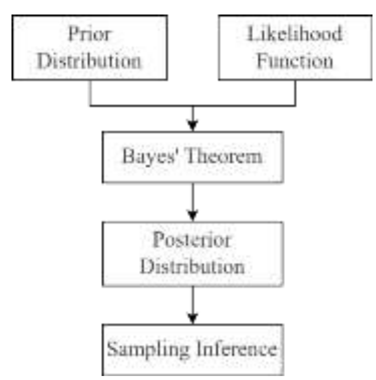


Figure 2. Parameter estimation process based on Bayesian theory

1) Specify the prior distribution: Determine the prior distribution $\pi(\theta)$ of the parameter θ based on its prior information;

2) Construct the likelihood function: Derive the likelihood function *f*(*x*|*θ*) for the sample $x=(x_1,\cdots,x_n)$, as shown in (16):

$$f(x|\ \theta)=\prod_{i=1}^{n} f\left(x_i|\ \theta\right) \tag{16}$$

3) Obtain the posterior distribution: Calculate the posterior probability density function of the parameter *θ* by combining the prior distribution *π(θ)* and the likelihood function *f(x|θ)* using Bayes' theorem.

4) Sampling and inference: Generate samples from the posterior distribution through MCMC or other sampling methods to perform parameter estimation and statistical inference.

The MCMC algorithm generates simulated samples of parameter vectors directly from the posterior distribution through numerical simulation. Common MCMC algorithms fall into two categories: The Metropolis-Hastings algorithm and the Gibbs sampling algorithm. The efficiency of the Metropolis-Hastings algorithm depends on the similarity between the chosen proposal density function and the true posterior density function. In contrast, the Gibbs sampling algorithm avoids this issue, as it imposes less stringent requirements on the proposal density function while achieving higher computational efficiency. In practical applications, the MCMC-Gibbs sampling process can be implemented using software packages such as OpenBUGS and WinBUGS.

The Gibbs sampling algorithm is implemented as follows; given the parameter vector $\theta=(\theta_1,\theta_2,\cdots,\theta_q)$ and observed data $data=(data_1,data_2,\cdots,data_n)$, with the posterior distribution $p(\theta_1|\ \theta_2,\cdots,\theta_q,data)$:

1) Assign initial values: Set an initial parameter vector $\theta^{(0)}=\left(\theta_1^{(0)},\theta_2^{(0)},\cdots,\theta_q^{(0)}\right)$ that conforms to the Markov chain properties.

2) Sample from conditional distributions: Sample $\theta_1^{(1)}$ from the conditional probability density $p\left(\theta_1|\ \theta_2^{(0)},\theta_3^{(0)},\cdots,\theta_q^{(0)},data\right)$;

3) Complete one iteration: Repeat 2) until $\theta_q^{(1)}$ is extracted from $p\left(\theta_q|\ \theta_1^{(0)},\theta_2^{(0)},\cdots,\theta_{q-1}^{(0)},data\right)$;

4) Iterate: Repeat steps 2)-3) for *m* iterations to obtain $\theta^{(m)}=\left(\theta_1^{(m)},\theta_2^{(m)},\cdots,\theta_q^{(m)}\right)$.

When *m* is sufficiently large, $\theta^{(m)}$ can be considered as samples drawn from the posterior *p(θ|data)*, enabling the estimation of unknown parameters.

B. Model superiority test method

For the degradation test data, when there are multiple degradation models to choose from, it is necessary to select the model that fits the degradation data the best, and Akaike information criterion (AIC) is usually used for the model optimization, and the likelihood ratio test is used to judge whether a simple model can be used in place of a complex model.

The AIC criterion is proposed by Akaike, a Japanese statistician, which evaluates the model fit in terms of the value of the likelihood function to measure the degree of fit and the number of unknown parameters in the model, which is the weighted value of the value of the likelihood function and the number of parameters. The smaller the value of the AIC, the better the model fit, and the formula is shown in (17):

$$AIC=2(\gamma-\ln(L(\theta\,|\,Y))) \tag{17}$$

where γ is the number of unknown parameters in the model and *L(θ|Y)* is the value of the likelihood function.

In order to visually compare and verify the reasonableness of the model, Quantile-Quantile(Q-Q) Plot is introduced as a visualization tool to assess the degree of fit between the sample data distribution and the theoretical distribution (e.g., normal distribution), as shown in Figure 3. which is a sample Q-Q Plot generated based on the sample of normal distribution, which can be used to illustrate how to assess the degree of fit between the sample data distribution and theoretical The degree of fit to the normal distribution. Data points close to the diagonal line indicate a good fit to the distribution, and in the Q-Q plot, the horizontal axis corresponds to the quantiles of the theoretical distribution, while the vertical axis shows the quantiles of the sample data.

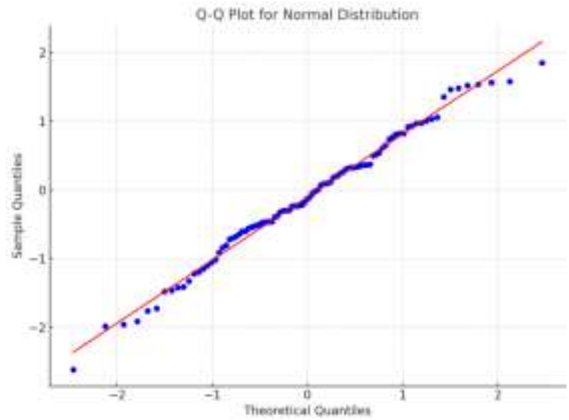


Figure 3. Example of a Q-Q plot

Ideally, the data points should be close to the diagonal distribution, indicating that the sample distribution matches the theoretical distribution; if they deviate, they reflect distributional differences, e.g., tail deviations may point to long-tailed distributions or extremes, whereas center deviations may indicate distributional skewness. The combination of Q-Q plots and AIC criterion can comprehensively assess the distributional hypotheses and provide strong support for the model fit goodness-of-fit tests.

V. EXPERIMENTS AND ANALYSIS OF RESULTS

A. Fiber Optic Gyroscope Performance Degradation Data

The following section integrates the fiber optic gyroscope's zero-bias degradation data with the simulation verification of the proposed method. The following figure shows the zero-bias performance degradation data of a fiber-optic gyroscope collected in the constant temperature stress accelerated degradation test, in which the constant temperature stress of the fiber-optic gyroscope is controlled to be 55℃, and three fiber-optic gyroscopes of the same model equipped in an inertial platform are selected. During the test, the performance degradation data of zero bias of each sample of fiber-optic gyroscope were tested and recorded over the course of 25 times, with a recording interval of 168h, and a total test time of 4032h, and no failure occurred. The curves of each performance parameter of the fiber optic gyroscope over time are obtained after the test, as shown in Figure 4.

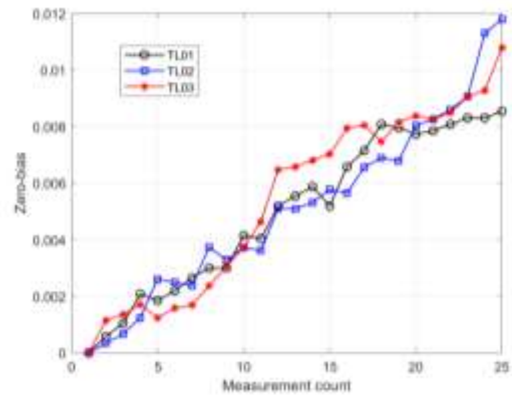


Figure 4. Zero-bias degradation data

Based on specific application requirements, the failure threshold was determined as 0.1 °/h. Over time, the zero-bias performance metrics of the three fiber optic gyroscopes (FOGs) exhibited a gradual increase, demonstrating an irreversible overall trend. This upward trend in values indicates a decline in FOG accuracy, reflecting cumulative damage under long-term thermal stress and characterizing the performance degradation of FOGs. Furthermore, the performance degradation data of the three FOG samples showed increased variability with prolonged time, highlighting growing inter-sample performance differences due to individual variations and stochastic factors.

The Wiener process model established in this paper describes a performance degradation process with nonlinearity and individual variability, which aligns with the observed characteristics of the degradation data. This consistency confirms the rationality of employing the Wiener process to model FOG performance degradation. For comparative validation of model feasibility and accuracy in reliability assessment, the standard Wiener process model is designated as M1, while the degenerate model with a nonlinear random effect Wiener process is denoted M2 in subsequent discussions.

B. Estimation of model parameters

1) Parameter estimation of the standard Wiener process model

First, under the standard Wiener process model M1, the prior distributions of the parameters are set as follows:

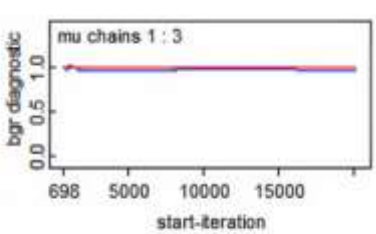
$$\mu \sim N(0,\, 0.0001)\,,\quad \sigma^{-2} \sim Gamma(0.001,\, 0.001)$$

In the OpenBUGS software, the MCMC-Gibbs sampling method was employed to generate three Markov chains. Each chain draws 40,000 samples from the posterior distribution to ensure convergence and robustness. TABLE I. lists the statistical measures of the parameter estimates for Model M1, including the mean and standard deviation. The use of multiple chains aims to assess model convergence and validate the stability of parameter estimates by comparing results across different chains.

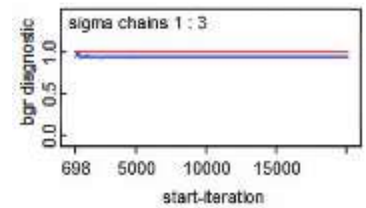
TABLE I. MCMC SAMPLING RESULTS FOR MODEL M1 PARAMETERS

Para-meter	Mean	Standard Deviation	MC Error	95% credible interval
μ	2.574E-6	4.022E-7	1.142E-9	(1.782E-6,3.376E-6)
σ	5.725E-4	4.868E-5	1.516E-7	(4.87E-4,6.776E-4)

The convergence diagnostics for the parameters of Model M1 are illustrated in Figure 5. The figure displays the results of the Gelman-Rubin diagnostic (BGR diagnostic), which was used to evaluate the convergence of parameters (μ and σ) across three independent MCMC chains. In the figure, all BGR values of the chains are close to 1, indicating minimal differences between chains and consistent within-chain and between-chain variances. This demonstrates that the MCMC sampling for parameters μ and σ has converged, and the simulation results are reliable.



a) The convergence diagnostic process for μ



b) The convergence diagnostic process for σ

Figure 5. Diagnostic results of convergence of model M1 parameters

2) Parameter Estimation of the Nonlinear Random Effects Wiener Process Model

Next, under the nonlinear random-effects Wiener process Model M2, the prior distributions of the parameters are specified as follows:

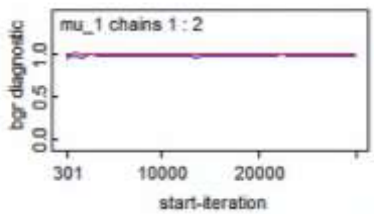
$$\mu_{\beta 1} \sim N(0,\, 0.0001)\,,\quad \sigma_{\beta 1}^{-2} \sim Gamma(0.001,\, 0.001)$$
$$\sigma_1^{-2} \sim Gamma(0.001,\, 0.001)\,,\quad a_1 \sim Uniform(0,\, 1)$$

In the OpenBUGS software, the MCMC-Gibbs sampling method was employed with random initial values assigned to parameters to generate two Markov chains. Each chain drew 40,000 samples from the posterior distribution to obtain parameter estimates. TABLE II. lists the mean, standard deviation, and other statistical measures of the parameter estimates for Model M2.

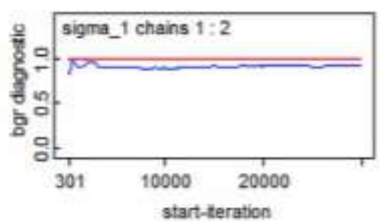
TABLE II. MCMC SAMPLING RESULTS FOR MODEL M2 PARAMETERS

Para-meter	Mean	Standard Deviation	MC Error	95% credible interval
$\mu_{\beta 1}$	6.79E-4	5.059E-5	2.316E-7	(5.815E-4,7.817E-4)
$\sigma_{\beta 1}$	4.455E-5	6.103E-6	3.656E-8	(2.793E-5,4.987E-5)
σ_1	8.551E-4	3.427E-5	1.356E-7	(7.732E-4,8.983E-4)
a_1	0.7467	0.003256	1.766E-5	(0.738,0.7499)

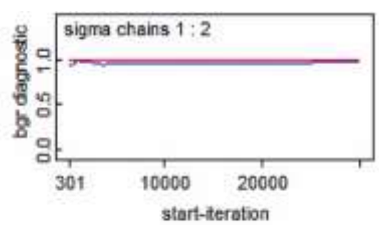
The convergence diagnostics for the parameters of Model M2 are illustrated in Figure 6. The figure displays the results of the Gelman-Rubin diagnostic (BGR diagnostic), which was used to evaluate the convergence of parameters $\mu_{\beta 1}$, $\sigma_{\beta 1}$, σ_1, a_1 across two independent MCMC chains.In the figure, all BGR values for the parameters rapidly approach and stabilize near the ideal value of 1.0, indicating consistent distributions across sampling chains and robust convergence. This demonstrates that the MCMC sampling for parameters $\mu_{\beta 1}$, $\sigma_{\beta 1}$, σ_1, a_1 has successfully converged, with no significant discrepancies in the sampling process. Consequently, the model results are reliable, and the posterior distributions of the parameters are credible.



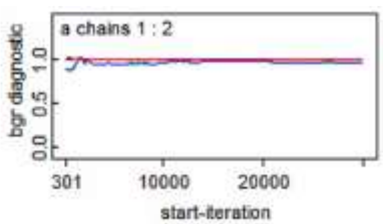
a) The convergence diagnostic process for $\mu_{\beta 1}$



b) The convergence diagnostic process for $\sigma_{\beta 1}$



c) The convergence diagnostic process for σ_1



d) The convergence diagnostic process for a

Figure 6. Diagnostic results of convergence of model M2 parameters

C. Goodness-of-Fit Test for Degradation Models

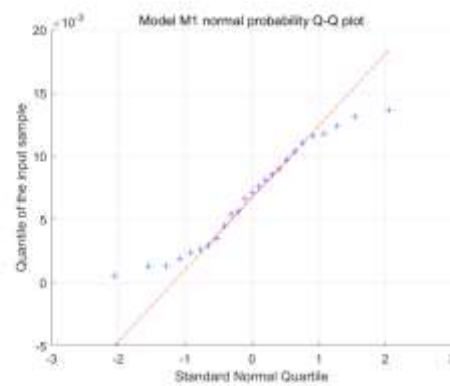
As shown in Figure 5. and Figure 6. , the parameters of both models M1 and M2 exhibit good convergence. By substituting the parameter estimates from TABLE I. and TABLE II. into (13) and (14), respectively, the log-likelihood values ln(L(θ)) and AIC values can be derived, as summarized in TABLE III. These metrics are used to compare the goodness-of-fit and complexity of the two models.

TABLE III. LOG-LIKELIHOOD AND AIC VALUES FOR DIFFERENT MODELS

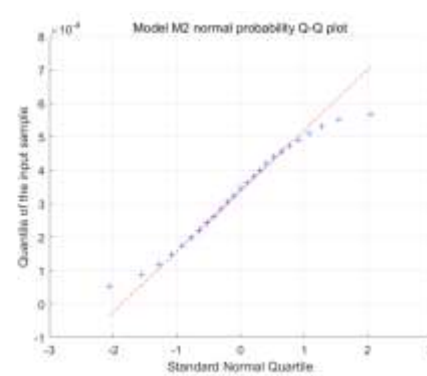
Model	**ln($L(\theta)$)**	**AIC**
M1	-286.1956	576.3911
M2	-201.5358	411.0715

From the comparison of M1 and M2 in TABLE III. , it is evident that model M2 exhibits a larger log-likelihood value and a smaller AIC value. This suggests that model M1 performs relatively poorly, as its ability to describe the data is either insufficient or overly complex. In contrast, model M2 provides a better fit for the data while simultaneously maintaining lower model complexity. Thus, model M2 is clearly superior to M1.

The Q-Q plots of models M1 and M2 are given in Figure 7. According to the definition of Q-Q plots, it can be seen that the stronger the linear relationship of Q-Q plots, the better the fit of the model.



a) Model M1



b) Model M2

Figure 7. Comparison results of Q-Q plots for M1 and M2

From an analysis of the Q-Q plots of the two models presented in Figure 7. , it can be seen that the data points in the Q-Q plot of Model M1 are generally distributed along the red dashed line (the theoretical quantile of the normal distribution), but the deviation is more pronounced, especially in the tail data on the left and the right. The deviation from the theoretical quantile in the tail data suggests that there may be some non-normality in the data for model M1. The data points fit better in the middle region, but the overall fit is poor. The Q-Q plot of model M2 has data points closer to the red dashed line, especially in the middle and tail regions, and the fit is significantly better than that of model M1. the smaller deviation of the tail data from the theoretical quantile suggests that the data distribution of model M2 is closer to the normal distribution. Overall, both models M1 and M2 can describe the degradation process of the fiber-optic gyroscope, which verifies the reasonableness of the modeling. Among them, the Q-Q plot of model M2 shows a stronger linear trend and the data points are closer to the theoretical normal distribution, so M2 is more suitable for describing the zero-bias degradation process of the fiber optic gyro.

D. Analysis of Reliability Assessment Results

Combined with the specific application conditions of the fiber optic gyro in the test, the failure threshold of the performance parameters is determined to be zero bias (≥0.1°/h). By substituting the estimated values of the parameters and the failure thresholds of the performance parameters of the models M1 and M2 into the (6) and (12), the zero-bias reliability curves of the fiber optic gyro's performance parameters based on the standard linear Wiener process and the zero-bias reliability curves of the fiber optic gyro's performance parameters based on the non-linear random effects of the Wiener process are obtained, which are shown in Figure 8.

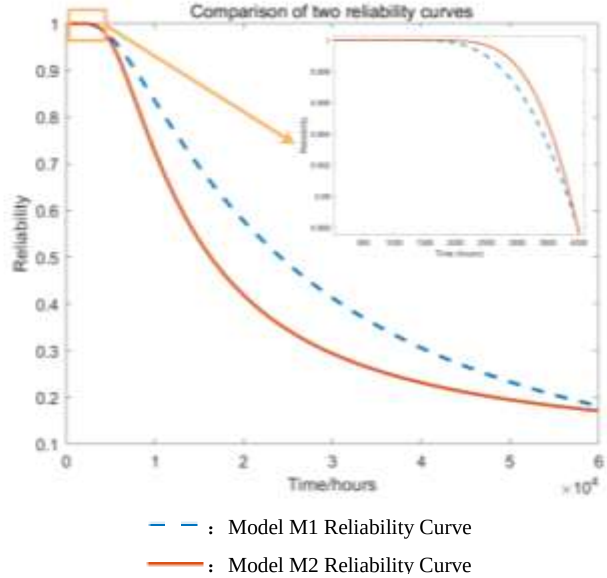


Figure 8. Model M1 vs. M2 Reliability Curve

As can be seen from Figure 8. , if the zero-bias performance degradation trajectory of the fiber-optic gyroscopes in this batch is described by model M1, the reliability is lower than 1 from 1,704 h. In fact, none of the three gyroscopes tested failed by the 4032h test cutoff. In contrast, model M2 is more consistent with the reliability characteristics of fiber-optic gyroscopes with reliability below 1 from 2105 h. The model M1 is more consistent with the reliability characteristics of fiber-optic gyroscopes. Meanwhile, model M1 is more optimistic because the reliability curve declines more slowly without considering individual differences, whereas model M2 is more conservative but closer to the reality because the reliability declines more rapidly compared to M1 after the introduction of the consideration of individual differences and nonlinearities. Combined with the zero-bias failure threshold (≥0.1°/h), M2 is more suitable for long-term use scenarios, providing a more rigorous basis for equipment maintenance and avoiding potential risks caused by performance degradation.

In conclusion, the M2 model stands out as the most effective model, as it effectively accounts for both individual differences and nonlinear behaviors in the degradation process of fiber-optic gyroscopes.

VI. CONCLUSIONS

In order to handle the individual variability and nonlinearity of performance degradation of fiber-optic gyros in reliability assessment, a method based on the nonlinear random effect Wiener process is proposed in this paper. The results show that the model can more accurately characterize the actual degradation behavior of the fiber optic gyro when considering the individual variability of the fiber optic gyro and the nonlinear characteristics of the performance degradation. In the validation of fiber-optic gyro zero-bias performance degradation data, the log-likelihood value of the proposed method is higher and the Akaike Information Criterion (AIC) value is lower than that of the model that only considers randomness, and the Q-Q plot exhibits characteristics that are closer to normal distribution, which proves the superior fitting ability of the model in interpreting the data. In conclusion, the model proposed in this paper can provide a more reasonable and accurate basis for the reliability assessment of fiber optic gyro and provides new theoretical support for the reliability analysis and performance optimization of inertial navigation equipment

REFERENCES

[1] Zhang, T.L., Wen, K.H., Zhang, X., Lin, C.F., Yang, J. Thermal-induced zero drift error mechanism and suppression method in fiber optic gyroscope light path. Journal of Projectiles, Rockets, Missiles and Guidance, 2024, 44(5): 14-24.

[2] Fu, J., Chang, Y., Ning, Z.W. Research progress in miniaturization technology of fiber optic gyroscopes. Sensors and Microsystems, 2020, 39(7): 1-4, 7.

[3] Wang, W. Fiber Optic Gyroscope Inertial Systems. Beijing: China Aerospace Publishing House, 2010.

[4] Liu, W.R., Li, D.C., Li, M.C., et al. Prospects for high-precision fiber optic gyroscope inertial navigation technology in maritime applications. Navigation and Control, 2022, 21(5): 241-249.

[5] Zhao, J.Y. Reliability Modeling and Application Research Based on Performance Degradation Data. National University of Defense Technology, 2005.

[6] Lin, M.Y.Z. Degradation-Based Product Fault Diagnosis and Health State Prediction Model Construction with Applications. Hunan University, 2023.

[7] Zhai, Y.L., Zhang, Z.H., Shao, S.S. Reliability modeling for products under multiple degradation mechanisms. Systems Engineering and Electronics, 2021, 43(6): 1714-1720.

[8] Wang, H.W., Teng, K.N. A review of reliability assessment techniques based on accelerated degradation data. Systems Engineering and Electronics, 2017, 39(12): 2877-2885.

[9] Jin, G. Degradation-Based Reliability Technology: Models, Methods, and Applications. Beijing: National Defense Industry Press, 2014.

[10] Qiu, R.H., Ju, K.L., Dong, Y.G., et al. Research on reliability testing of small-sample electric spindles based on performance degradation. China Mechanical Engineering, 2016, 27(20): 2738-2742+2748.

[11] Wang, H.W., Xi, W.J., Zhao, J.Y., et al. Reliability evaluation method based on degradation distribution under accelerated stress. Systems Engineering and Electronics, 2016, 38(1): 239-244.

[12] Zhang Z, Si X, Hu C, et al. Degradation data analysis and remaining useful life estimation: A review on Wiener-process-based methods [J]. European Journal of Operational Research, 2018, 271(3): 775-796.

[13] Fouladirad M, Giorgio M, Pulcini G. A transformed gamma process for bounded degradation phenomena [J]. Quality and Reliability Engineering International,2023,39(2):546-564.

[14] Wang, S.T. Design method for step-stress accelerated degradation testing based on bivariate inverse Gaussian process. Tianjin: Tianjin University, 2020.

[15] Liu D,Wang S, Cui X. An artificial neural network supported Wiener process based reliability estimation method considering individual difference and measurement error [J]. Reliability Engineering and System Safety,2022,218:108162.

[16] Wang, N.C., Hu, D.Y., Liu, Q., et al. Reliability analysis of turbine disk tenon groove crack propagation based on Wiener process. Journal of Aerospace Power, 2022, 37(11): 2440–2447.

[17] Zhu Y, Liu S, Wei K, et al. A novel based-performance degradation Wiener process model for real-time reliability evaluation of lithium-ion battery [J]. Journal of Energy Storage,2022,50:104313.

[18] LEFEVRE H. The fiber-optic gyroscope. 2nd ed. Boston,USA: Artech House, 2014.