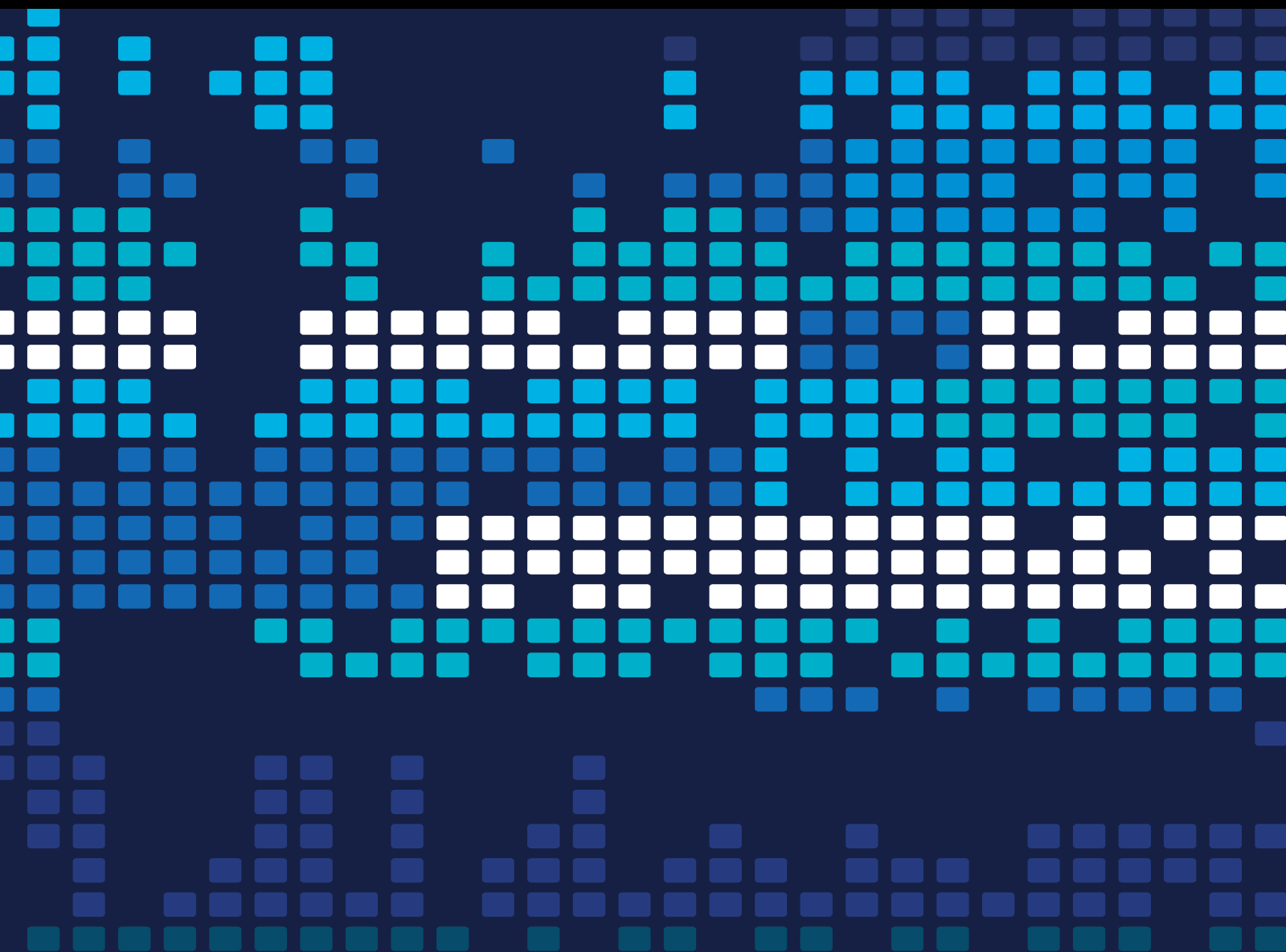


Methodologies, Algorithms, and Applications of Artificial Intelligence and Internet of Things

Lead Guest Editor: Ting Yang

Guest Editors: Qiang Yang, Javid Taheri, and Wei Li





Methodologies, Algorithms, and Applications of Artificial Intelligence and Internet of Things

Scientific Programming

**Methodologies, Algorithms, and
Applications of Artificial Intelligence
and Internet of Things**

Lead Guest Editor: Ting Yang


Guest Editors: Qiang Yang, Javid Taheri, and Wei Li



Copyright © 2021 Hindawi Limited. All rights reserved.

This is a special issue published in "Scientific Programming." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Chief Editor


Emiliano Tramontana , Italy

Academic Editors

Marco Aldinucci , Italy
Daniela Briola, Italy
Debo Cheng , Australia
Ferruccio Damiani , Italy
Sergio Di Martino , Italy
Sheng Du , China
Basilio B. Fragueta , Spain
Jianping Gou , China
Jiwei Huang , China
Sadiq Hussain , India
Shujuan Jiang , China
Oscar Karnalim, Indonesia
José E. Labra, Spain
Maurizio Leotta , Italy
Zhihan Liu , China
Piotr Luszczek, USA
Tomàs Margalef , Spain
Cristian Mateos , Argentina
Zahid Mehmood , Pakistan
Roberto Natella , Italy
Diego Oliva, Mexico
Antonio J. Peña , Spain
Danilo Pianini , Italy
Jiangbo Qian , China
David Ruano-Ordás , Spain
Željko Stević , Bosnia and Herzegovina
Kangkang Sun , China
Zhiri Tang , Hong Kong
Autilia Vitiello , Italy
Pengwei Wang , China
Jan Weglarz, Poland
Hong Wenxing , China
Dongpo Xu , China
Tolga Zaman, Turkey







Contents

Normalized Combinations of Proportionate Affine Projection Sign Subband Adaptive Filter

Tong An, Tao Zhang, Yanzhang Geng , and Haiquan Jiao




Research Article (12 pages), Article ID 8826868, Volume 2021 (2021)

IOT-Based Cotton Whitefly Prediction Using Deep Learning

Rana Muhammad Saleem , Rafaqat Kazmi , Imran Sarwar Bajwa , Amna Ashraf , Shabana Ramzan , and Waheed Anwar 



Research Article (17 pages), Article ID 8824601, Volume 2021 (2021)

Clone Chaotic Parallel Evolutionary Algorithm for Low-Energy Clustering in High-Density Wireless Sensor Networks

Rui Yang , Mengying Xu , and Jie Zhou 

Research Article (13 pages), Article ID 6630322, Volume 2021 (2021)

ICS Software Trust Measurement Method Based on Dynamic Length Trust Chain

Wenli Shang  and Xiangyu Xing 





Research Article (11 pages), Article ID 6691696, Volume 2021 (2021)

A Test Cases Generation Method for Industrial Control Protocol Test

Wenli Shang , Guanyu Zhang , Tianyu Wang , and Rui Zhang 



Research Article (9 pages), Article ID 6611732, Volume 2021 (2021)

An Algorithm of Occlusion Detection for the Surveillance Camera

Peng Shi , Bin Hou , Jing Chen , and Yunxiao Zu 



Research Article (9 pages), Article ID 6698160, Volume 2021 (2021)

Distribution Network Topology Identification Based on IEC 61850 Logical Nodes

Yu Chen , Lingyan Sun , Zonghui Wang, and Jinghua Wang





Research Article (8 pages), Article ID 6639432, Volume 2021 (2021)

Research on Distributed Feeder Automation Communication Based on XMPP and GOOSE

Lingyan Sun , Yu Chen , Chuiyue Kong, and Jinghua Wang


Research Article (11 pages), Article ID 6650725, Volume 2021 (2021)

Adaptive Particle Swarm Optimization with Gaussian Perturbation and Mutation

Binbin Chen , Rui Zhang , Long Chen , and Shengjie Long 

Research Article (14 pages), Article ID 6676449, Volume 2021 (2021)

Continuous Trust Evaluation of Power Equipment and Users Based on Risk Measurement


Congcong Shi , Jiaxuan Fei, Xiaojian Zhang, Qigui Yao, and Jie Fan

Research Article (6 pages), Article ID 8895804, Volume 2020 (2020)


Intelligent Detection and Recovery of Missing Electric Load Data Based on Cascaded Convolutional Autoencoders

Xin Wang, Yuanyi Chen , Wei Ruan , Qiang Gao, Guode Ying, and Li Dong

Research Article (20 pages), Article ID 8828745, Volume 2020 (2020)



The Abnormal Detection for Network Traffic of Power IoT Based on Device Portrait

Jiaxuan Fei , Qigui Yao , Mingliang Chen , Xiangqun Wang , and Jie Fan 

Research Article (9 pages), Article ID 8872482, Volume 2020 (2020)

Research Article

Normalized Combinations of Proportionate Affine Projection Sign Subband Adaptive Filter

Tong An, Tao Zhang, Yanzhang Geng , and Haiquan Jiao

School of Electrical Information Engineering, Tianjin University, Tianjin 300072, China

Correspondence should be addressed to Yanzhang Geng; gregory@tju.edu.cn

Received 21 July 2020; Revised 16 September 2020; Accepted 18 August 2021; Published 27 August 2021

Academic Editor: Shah Nazir

Copyright © 2021 Tong An et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The proportionate affine projection sign subband adaptive filter (PAP-SSAF) has a better performance than the affine projection sign subband adaptive filter (AP-SSAF) when we eliminate the echoes. Still, the robustness of the PAP-SSAF algorithm is insufficient under unknown environmental conditions. Besides, the best balance remains to be found between low steady-state misalignment and fast convergence rate. In order to solve this problem, we propose a normalized combination of PAP-SSAF (NCPAP-SSAF) based on the normalized adaptation schema. In this paper, a power normalization adaptive rule for mixing parameters is proposed to further improve the performance of the NCPAP-SSAF algorithm. By using Nesterov's accelerated gradient (NAG) method, the mixing parameter of the control combination can be obtained with less time consumed when we take the l_1 -norm of the subband error as the cost function. We also test the algorithmic complexity and memory requirements to illustrate the rationality of our method. In brief, our study contributes a novel adaptive filter algorithm, accelerating the convergence speed, reducing the steady-state error, and improving the robustness. Thus, the proposed method can be utilized to improve the performance of echo cancellation. We will optimize the combination structure and simplify unnecessary calculations to reduce the algorithm's computational complexity in future research.

1. Introduction

Adaptive filters are important components of many signal processing applications, such as echo cancellation, system identification, channel equalization, and so on [1, 2]. Echo cancellation is the process of extracting pure signals from echo corrupted signals. The adaptive filter for the echo cancellation system is generally designed in the frequency domain. Otherwise, the length of the designed filter tap might be unexpected in the time domain. The prominent of the least mean squares (LMS), normalized LMS(NLMS), and Filtered-x LMS(FxLMS) are simple and reliable [3, 4]. However, for colored inputs, their performance will degrade, especially for the speech input signals [5]. Exploiting the multiple regression, the affine projection algorithm (APA) can improve convergence performance but at the cost of high computational complexity. Besides, the normalized subband adaptive filter (NSAF) could also speed up the convergence [6]. This kind of algorithm is presented from

the principle of minimum disturbance, and it processes the colored input signals by analyzing filter banks [7]. Das and Trivedi proved that the rate of convergence can also be improved by using the proportional normalization method in the adaptive filter. However, the sparseness of impulse response still impacts its performance [8].

In real life, the noise is complex and does not meet the Gaussian distribution, and many adaptive algorithms suffer reduced convergence rate under the impulsive noise environment due to the l_2 -norm optimization criterion [9]. The sign-algorithms family has the ability to resist the impulse noise disturbance. However, the convergence speed of SSAF is very slow and cannot be accelerated by increasing the number of subbands. In addition, if the impulse response of the echo path is sparse, the convergence speed of SSAF will further decrease. In order to speed up the convergence rate of the algorithm, Ni, J et al. proposed variable regularization parameter SSAF (VRP-SSAF) to further improve performance [10, 11].

In recent years, researchers proposed many modified SSAF have to accelerate the convergence. By adopting the idea of multiple regression, the APA algorithm shows better performance, which brings new inspiration to further studies. Reference [7] proposes an AP-SSAF algorithm that uses multiple previous input vectors to update the weight vector. Yu and Zhao discovered a phenomenon that the filter performance would decrease when all the subbands use the same common weighting factor [12]. To solve this problem, they proposed the method called the individual-weighting-factor SSAF (IWF-SSAF), which allocates an individual weighting factor for each subband. However, in many situations, the echo path impulse responses are sparse, so that these above-modified algorithms converge slowly [13]. For the sparse echo paths, these following mentioned algorithms can incorporate a gain distribution matrix into their adaptations. Consequently, considering the sparsity of the impulse responses, the SSAF, AP-SSAF, and IWF-SSAF were improved to P-SSAF, PAP-SSAF, and IWF-IP-SSAF, respectively [14–17].

Now we know that because of using the fixed step size, the standard SSAF algorithm and the modified SSAF algorithm should find the best point between fast convergence rate and low steady-state misalignment. To solve this problem, many variable step-size algorithms have been proposed [18–20], but all these algorithms need to incorporate the a priori information into the learning mechanism. However, it is difficult to obtain them from the real world.

In addition to the variable step size algorithm mentioned above, the combinatorial method, which combines two different step size filters by using mixing parameters, keeps a balance of optimal performance between the convergence rate and the steady-state error [21]. And it is also called convex combination because the mixing parameter ranges between 0 and 1. This algorithm uses a random gradient descent algorithm to determine the optimal solution. The improved convex combination normalized subband adaptive filter (ICNSAF) can achieve the desired performance without the information of the subband noise power. Considering the impulse noise, Lu et al. proposed a novel combination approach of the AP-SSAF, which uses weight transfer of coefficients to obtain fast convergence speed during the transition stage [22]. By applying the convex combination scheme to IWF-SSAF and cyclically returning the weight vector of the combined filter to both component filters, Yu et al. proposed the combined IWF-SSAF with weight feedback. Although the above-proposed combination algorithms improve the performance of adaptive filters to some extent, there are still two problems to be solved. First, the above algorithms do not achieve good adaptability in terms of mixing parameter step size, which is a major factor affecting the adaptability of the filter. To correctly adjust the step size of the mixing parameter, we also need to consider some characteristics of the filtering scheme, such as input signal and additive noise power, or the step size of the adaptive filter included in the combination [23]. Second,

they rarely considered the sparsity of impulse response, resulting in weak robustness of the filters in this situation [24, 25].

In this paper, the normalized combination of PAP-SSAF (NCPAP-SSAF) was proposed to deal with these defects, which adjusts the mixing parameter by means of the power normalization. Compared with other adaptive filter algorithms, the algorithm NCPAP-SSAF we proposed is robust in impulse noise environment. which is confirmed in the simulation results. In contrast to the standard PAP-SSAF algorithm, the proposed algorithm has the following characteristics:

- (i) In order to accelerate convergence and improve robustness against impulse noise, the l_1 -norm of subband error is used as the cost function in this paper. Then the mixing parameter of the combination is obtained by using a Nesterov's accelerated gradient (NAG) method.
- (ii) NCPAP-SSAF algorithm normalizes the step size of the mixing parameter so that it is independent of the signal-to-noise ratio (SNR). The improvement makes the adaptive filter easy to select the step size and shows a robust behavior against unknown environmental conditions such as the "double-talk" scene.

2. Background of PAP-SSAF

The algorithm we proposed in this paper is an optimizing method based on the PAP-SSAF algorithm in convergence rate and steady-state error. Therefore, it is necessary to introduce this algorithm first.

At the beginning, we analyze the mathematical model parameters of a typical echo canceller. The input signal vector $u(n)$ is filtered through the unknown impulse response $\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{L-1}(n)]^T$ to observe the desired signal, where L is the length of the impulse response. This process can be described as follows: $\mathbf{d}(n) = \mathbf{u}^T(n)\mathbf{w}(n) + \mathbf{v}(n)$, where $\mathbf{v}(n)$ represents the background noise, and superscript T represents transpose of matrix and $\mathbf{u}(n) = [u(n), u(n-1), \dots, u(n-L+1)]^T$. We define N as the number of subbands, $\mathbf{d}(n)$ as the microphone signal, and $\mathbf{u}(n)$ as the far-end signal in the described adaptive filter structure. First, by means of the analysis filters, $\mathbf{d}(n)$ and $\mathbf{u}(n)$ are stripped into N subband signals as $d_i(n)$ and $u_i(n)$, in which $i = 0, 1, \dots, N-1$. After the subband input signal $u_j(n)$ passing through the adaptive filter $\widehat{\mathbf{W}}(\mathbf{z}, \mathbf{k})$, we can get the subband output signal $y_i(n)$. The letter n represents the original sequence; k represents index decimated sequences. The results obtained from N -decimation of the filter are $d_{i,D}(k)$ and $y_{i,D}(k)$. The decimated subband error signal can be expressed as follows: $\mathbf{e}_{i,D}(\mathbf{k}) = \mathbf{d}_{i,D}(\mathbf{k}) - \mathbf{u}_i(\mathbf{k})\widehat{\mathbf{w}}(\mathbf{k})$, where $\widehat{\mathbf{w}}(\mathbf{k})$ is the tap-weight vector of the adaptive filter $\widehat{\mathbf{W}}(\mathbf{z}, \mathbf{k})$. After that, we use μ to represent the step length of the filter. Then, we can formulate the update of the SSAF as in the following equation:

$$\widehat{\mathbf{w}}(k+1) = \widehat{\mathbf{w}}(k) + \mu \frac{\mathbf{U}_A(k) \text{sgn}[\boldsymbol{\varepsilon}_A(k)]}{\sqrt{\{\mathbf{U}_A(k) \text{sgn}[\boldsymbol{\varepsilon}_A(k)]\}^T \{\mathbf{U}_A(k) \text{sgn}[\boldsymbol{\varepsilon}_A(k)]\} + \delta}}, \quad (1)$$

where $\boldsymbol{\varepsilon}_A$ represents posteriori subband error, and $\text{sgn}[\cdot]$ is the sign function. δ is the regularization factor which is a small constant to avoid numerator divided by zero. Inspired by the APA, Ni et al. in [7] proposed a method that used several previous input vectors to update the tap-weight vector, which they called it AP-SSAF. In each subband, we collect the nearest L -th desired subband signals to generate the i -th desired subband signal vector. Similarly, we collect the subband input vectors to generate the input signal matrix.

$$\begin{aligned} \mathbf{d}_i(k) &= [d_{i,D}(k), d_{i,D}(k-1), \dots, d_{i,D}(k-L+1)]^T, \\ \mathbf{U}_i(k) &= [\mathbf{u}_i(k), \mathbf{u}_i(k-1), \dots, \mathbf{u}_i(k-L+1)]. \end{aligned} \quad (2)$$

In AP-SSAF, it is necessary to obtain the prior subband error and the posterior subband error. Prior error is referred to as in the following equation:

$$\mathbf{e}_A(k) = \mathbf{d}_A(k) - \mathbf{U}_A^T(k) \widehat{\mathbf{w}}(k). \quad (3)$$

Posteriori error is referred to as in the following equation:

$$\boldsymbol{\varepsilon}_A(k) = \mathbf{d}_A(k) - \mathbf{U}_A^T(k) \widehat{\mathbf{w}}(k+1). \quad (4)$$

where,

$$\begin{aligned} \mathbf{d}_A(k) &= [\mathbf{d}_0^T(k), \mathbf{d}_1^T(k), \dots, \mathbf{d}_{N-1}^T(k)]^T, \\ \mathbf{U}_A(k) &= [\mathbf{U}_0(k), \mathbf{U}_1(k), \dots, \mathbf{U}_{N-1}(k)]. \end{aligned} \quad (5)$$

To formulate the AP-SSAF, it should follow constrained optimization problem as in the following equation:

$$\min_{\widehat{\mathbf{w}}(k+1)} \|\mathbf{d}_A(k) - \mathbf{U}_A^T(k) \widehat{\mathbf{w}}(k+1)\|_1, \quad (6)$$

$$\text{subject to } \|\widehat{\mathbf{w}}(k+1) - \widehat{\mathbf{w}}(k)\|_2^2 \leq \mu^2, \quad (7)$$

where $\|\cdot\|_1$ and $\|\cdot\|_2$ represent the l_1 -norm and l_2 -norm, respectively. By using the Lagrange multiplier method, the unconstrained optimization problem can be used to replace the above-constrained optimization problem, that is to say that we use the subband error vector \mathbf{e}_A instead of the posteriori subband error vector $\boldsymbol{\varepsilon}_A$ in equation (1). Accordingly, the renewal equation of AP-SSAF is as follows:

$$\widehat{\mathbf{w}}(k+1) = \widehat{\mathbf{w}}(k) + \mu \frac{\mathbf{U}_A(k) \text{sgn}[\mathbf{e}_A(k)]}{\sqrt{\{\mathbf{U}_A(k) \text{sgn}[\mathbf{e}_A(k)]\}^T \{\mathbf{U}_A(k) \text{sgn}[\mathbf{e}_A(k)]\} + \delta}} \quad (8)$$

In the real world, there is the fact that the echo path impulse response is usually sparse and most of the filter coefficients are extremely close to zero. To solve this problem, the work in [26] combined the proportionate idea with the AP-SSAF, which is called the PAP-SSAF, and the updated equation of the PAP-SSAF is as follows:

$$\widehat{\mathbf{w}}(k+1) = \widehat{\mathbf{w}}(k) + \mu \frac{\mathbf{G}(k) \mathbf{U}_A(k) \text{sgn}[\mathbf{e}_A(k)]}{\sqrt{\|\mathbf{G}(k) \mathbf{U}_A(k) \text{sgn}[\mathbf{e}_A(k)]\|_2^2 + \delta}} \quad (9)$$

In equation (9), $\mathbf{G}(k) = \text{diag}[g_1(k), g_2(k), \dots, g_k(k)]$ is a proportionate diagonal matrix. There are many algorithms that have been proposed to calculate the diagonal matrix [27]. Among them, a typical method shows robustness in the condition of the impulse response. This method has been used in the PAP-SSAF, and it will be used in our research as well. The diagonal elements from $\mathbf{G}(k)$ are calculated by the following:

$$g_m(n) = \frac{1-\beta}{2M} + (1+\beta) \frac{|\widehat{w}_m(n)|}{\|\widehat{\mathbf{w}}(n)\|_1 + \varepsilon}, \quad m = 1, 2, \dots, M. \quad (10)$$

3. Proposed NCPAP-SSAF

3.1. The Algorithm Design of NCPAP-SSAF. The step size has a great influence on the convergence performance. In terms of the adaptive filter, on the one hand, if the step size is large, the adaptive filter convergence is very fast, but it will lead to a larger steady-state error. On the other hand, the step size is small and then the convergence is slow, but there is a small steady-state error. The basic principle of the convex combination algorithm is combining two filters with different step sizes, which update independently. Consequently, the final filter inherits the advantages of the two filters.

For simplicity, we show one of all the subband structures of the convex combination method in Figure 1 $\widehat{\mathbf{w}}_1(\mathbf{k})$ denotes the filter vector with a large step size and $\widehat{\mathbf{w}}_2(\mathbf{k})$ denotes that with a small step size. The subband output signal of each filter is $\mathbf{y}_{i,D,j}(\mathbf{k}) = \widehat{\mathbf{w}}_j^T(\mathbf{k}) \mathbf{u}_j(\mathbf{n})$ and the subband error is $e_{i,D,j}(k) = d_{i,D}(k) - y_{i,D,j}(k)$, where $i=0, 1, \dots, N-1, j=1, 2$. In the combined filter structure, the two filters do not affect each other and update independently. They update according to the following:

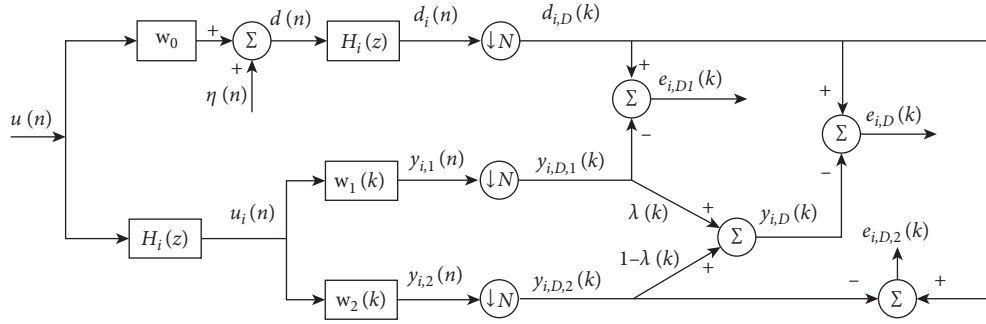


FIGURE 1: Structure of one subband in the proposed filter.

$$\hat{\mathbf{w}}_j(k+1) = \hat{\mathbf{w}}_j(k) + \mu_j \frac{\mathbf{G}_j(k) \mathbf{U}_A(k) \text{sgn}[\mathbf{e}_{A,j}(k)]}{\sqrt{\|\mathbf{G}_j(k) \mathbf{U}_A(k) \text{sgn}[\mathbf{e}_{A,j}(k)]\|_2^2 + \delta}}, \quad (11)$$

$$\mathbf{e}_{A,j}(k) = \mathbf{d}_A(k) - \mathbf{U}_A^T(k) \hat{\mathbf{w}}_j(k), \quad j = 1, 2. \quad (12)$$

We obtain the final output by combining the subband output of two filters, as follows:

$$y_{i,D}(k) = \lambda(k) y_{i,D,1}(k) + [1 - \lambda(k)] y_{i,D,2}(k), \quad (13)$$

where $\lambda(k)$ is the mixing parameter. Thus, the overall error can be expressed as follows:

$$\mathbf{e}_A(k) = \lambda(k) \mathbf{e}_{A,1}(k) + [1 - \lambda(k)] \mathbf{e}_{A,2}(k). \quad (14)$$

The weight vector of the combination filter is referred to as follows:

$$\hat{\mathbf{w}}(k) = \lambda(k) \hat{\mathbf{w}}_1(k) + [1 - \lambda(k)] \hat{\mathbf{w}}_2(k). \quad (15)$$

Since the value of $\lambda(k) \in [0, 1]$, this kind of combination method is named as convex combination and has usually been utilized in combinational filters. $\lambda(k)$ is calculated by the sigmoid function as follows:

$$\lambda(k) = \frac{1}{(1 + e^{-\alpha(k)})}. \quad (16)$$

The main problem in designing a convex composite filter is how to find the appropriate value of $\alpha(k)$ to make the mean square error of the error signal minimized. For a lossless filter bank, the power of the output error is equal to the sum of the powers of the subband errors [28]. The traditional convex combination algorithm uses the stochastic gradient method to determine the mixing parameter. However, its convergence performance is unsatisfactory, so that the filter cannot track the system quickly. In order to solve this problem and improve the capability of impulse noise suppression, we use the Nesterov's accelerated gradient (NAG) method to determine the mixing parameter.

The key point of the Nesterov's accelerated gradient algorithm is illustrated as shown in Figure 2. NAG can be unfolded into two steps: Firstly, we calculate the update vector $\alpha(k)$ according to the past time step $\alpha(k-1)$ and the gradient $\nabla_\alpha J(k)$ obtained from the next position of the

parameters. Note that computing $J(k)$ gives us a rough idea of where our parameters are going to be. And we can look "ahead" by calculating the gradient not w.r.t. to the current parameters $\alpha(k-1)$ but w.r.t. the approximate future position of the parameters. Finally, we update the parameter $\alpha(k)$ and accomplish this iteration. Therefore, it will not stop convergence before beyond the region of local optimal solution.

$$J(k) = \sum_{i=0}^{N-1} |e_{i,D}(k)|. \quad (17)$$

This method can minimize cost function as equation (19) [29]:

The update equation for $\alpha(k)$ is given by the following equation:

$$\alpha(k) = \alpha(k-1) - v_k, \quad (18)$$

$$v_k = \gamma v_{k-1} - \mu_\alpha \nabla_\alpha J(k), \quad (19)$$

$$\nabla_\alpha J(k) = -\lambda(k) [1 - \lambda(k)] \sum_{i=0}^{N-1} \text{sgn}(e_{i,D}(k)) \cdot [y_{i,D,1}(k) - y_{i,D,2}(k)], \quad (20)$$

where v_k is the momentum, γ is a constant number named momentum factor, and it ranges between zero and one, and $\mu_\alpha > 0$ is the step size. It can be seen that the NAG is the same as the stochastic gradient method when $\gamma = 0$. Reference [16] points out that $\alpha(k)$ is limited to a symmetrical interval $[-\alpha^+, \alpha^+]$ to meet the minimum level of adaptation. The experimental results in [16] indicate that the optimal value of α^+ should be set to 4, while the value of $\lambda(k)$ is restricted in the range of [0.018 0.982].

3.2. Power Normalized Rule for Adapting the Mixing Parameter. When the value of the mixing parameter step μ_α is reasonable, the update equation of $\alpha(k)$ can provide good performance for the whole system. However, the value of μ_α is related to many factors of the filter, such as input signals and additive noise power and the step size of the adaptive filter. Therefore, it is necessary to normalize the mixing parameter step μ_α . Substituting equations (19) and (20) into the update equation (18), we get the following:

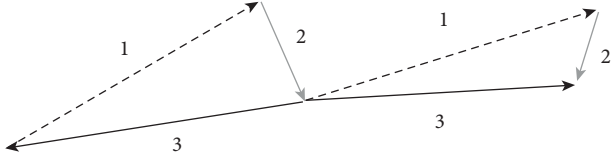


FIGURE 2: The NAG algorithm. The dotted line 1 represents the past time step $\alpha(k-1)$ and the grey line 2 represents the gradient $\nabla_{\alpha} J(k)$ obtained from the next position of the parameters. The update vector $\alpha(k)$ is indicated by the black solid line 3.

$$\alpha(k) = \alpha(k-1) - \gamma v_{t-1} - \mu_{\alpha} \lambda(k) [1 - \lambda(k)] \cdot [\mathbf{y}_{D,1}(k) - \mathbf{y}_{D,2}(k)]^T \text{sgn}[\mathbf{e}_D(k)]. \quad (21)$$

In equation (21), $\mathbf{e}_D(k) = [e_{0,D}(k), e_{1,D}(k), \dots, e_{N-1,D}(k)]^T$ denotes the overall error vector, and $\mathbf{y}_{D,j}(k) = [y_{0,D,j}(k), y_{1,D,j}(k), \dots, y_{N-1,D,j}(k)]^T$ ($j = 1, 2$) denotes the output of each filter.

Under the condition that $N = 1$, the filter can be seen as a full band filter, and the signal has no use for analysis and reconstruction. The adaptive rule of mixing parameter is equivalent to the sign-error-LMS algorithm, where $\mu_{\alpha} \lambda(k) [1 - \lambda(k)]$ is the varying step size and the input signal is $[e_{0,D,1}(k) - e_{0,D,2}(k)]$. The reason for this equivalent analysis is that the output of the combinational filter can be expressed as follows:

$$\begin{aligned} y_{0,D}(k) &= \lambda(k) y_{0,D,1}(k) + [1 - \lambda(k)] y_{0,D,2}(k) \\ &= y_{0,D,2}(k) + \lambda(k) [y_{0,D,1}(k) - y_{0,D,2}(k)] \\ &= y_{0,D,2}(k) + \lambda(k) [e_{0,D,1}(k) - e_{0,D,2}(k)]. \end{aligned} \quad (22)$$

So the overall combination scheme can be seen as a two-layer adaptive filter. According to their own rules, the two-component filters operate independently in the first layer. In the second layer, the output error of the two combination filters in the first layer is taken as input so that the norm of the overall output error is minimized. Since $[e_{0,D,1}(k) - e_{0,D,2}(k)]$ is the input signal at this level, it makes sense to use the above adaptive scheme. Reference [30] proved that the system performance of ε -normalized-SLMS (NSLMS) is better than SLMS. According to this conclusion, if we replace SLMS with NSLMS in the update of $\alpha(k)$, the system performance will be further improved. After normalized, the update equation of $\alpha(k)$ is referred to as follows:

$$\alpha(k) = \alpha(k-1) - \gamma v_{t-1} - \mu_{\alpha} \lambda(k) [1 - \lambda(k)] \cdot \frac{e_{0,D,1}(k) - e_{0,D,2}(k)}{[e_{0,D,1}(k) - e_{0,D,2}(k)]^2} \text{sgn}[e_{0,D}(k)]. \quad (23)$$

However, the instantaneous value $[e_{0,D,1}(k) - e_{0,D,2}(k)]^2$ is an inaccurate estimation of the input signal, so the calculation is not stable. The algorithm can be improved effectively when the power estimation of the input signal is used instead of its instantaneous value, as equation (24). When $N = 1$, the filter is a full band filter,

the instantaneous value can be replaced by its power estimation:

$$\alpha(k) = \alpha(k-1) - \gamma v_{t-1} - \mu_{\alpha} \lambda(k) [1 - \lambda(k)] \cdot \frac{e_{0,D,1}(k) - e_{0,D,2}(k)}{\hat{\sigma}_p^2(k)} \text{sgn}[e_{0,D}(k)], \quad (24)$$

$$\hat{\sigma}_p^2(k) = \eta \hat{\sigma}_p^2(k-1) + (1 - \eta) [e_{0,D,1} - e_{0,D,2}]^2, \quad (25)$$

where η named forgetting factor is close to 1, such as 0.99. When $N > 1$, the update rule of the mixing parameter is the same as the SSAF. Its step size is $\mu_{\alpha} \lambda(k) [1 - \lambda(k)]$ and its subband input signal is $\mathbf{e}_{D,1}(k) - \mathbf{e}_{D,2}(k) = [e_{0,D,1}(k) - e_{0,D,2}(k), e_{1,D,1}(k) - e_{1,D,2}(k), \dots, e_{N-1,D,1}(k) - e_{N-1,D,2}(k)]$. The output of the combined filter can be expressed as follows:

$$\begin{aligned} \mathbf{y}_D(k) &= \mathbf{y}_{D,2}(k) + \lambda(k) [\mathbf{y}_{D,1}(k) - \mathbf{y}_{D,2}(k)] \\ &= \mathbf{y}_{D,2}(k) + \lambda(k) [\mathbf{e}_{D,1}(k) - \mathbf{e}_{D,2}(k)], \end{aligned} \quad (26)$$

which supports the above conclusions. In (26) $\mathbf{y}_D(k) = [y_{0,D}(k), y_{1,D}(k), \dots, y_{N-1,D}(k)]^T$. Then comparing the update equation (21) with the standard SSAF, we can easily see that the step size of the former has not been normalized. Therefore, normalizing the step size of $\alpha(k)$ can improve the convergence performance of the global filter. By analogizing the updating method of filter weights in SSAF, we can get the normalized expression of $\alpha(k)$ step size as follows:

$$\alpha(k) = \alpha(k-1) - \gamma v_{t-1} - \mu_{\alpha} \lambda(k) [1 - \lambda(k)] \cdot \frac{[\mathbf{e}_{D,1}(k) - \mathbf{e}_{D,2}(k)]^T}{\sqrt{\sum_{i=0}^{N-1} [e_{i,D,1}(k) - e_{i,D,2}(k)]^2}} \text{sgn}[\mathbf{e}_D(k)]. \quad (27)$$

Similar to the condition when $N = 1$, the instantaneous value $[e_{0,D,1}(k) - e_{0,D,2}(k)]^2$ cannot be used to estimate the power of the second layer input signal very well, and a better behavior is obtained from the following equation:

$$\alpha(k) = \alpha(k-1) - \gamma v_{t-1} - \mu_{\alpha} \lambda(k) [1 - \lambda(k)] \cdot \frac{[\mathbf{e}_{D,1}(k) - \mathbf{e}_{D,2}(k)]^T}{\hat{\sigma}_p(k)} \text{sgn}[\mathbf{e}_D(k)], \quad (28)$$

$$\hat{\sigma}_p(k) = \eta \hat{\sigma}_p(k-1) + (1 - \lambda) \sqrt{\sum_{i=0}^{N-1} [e_{i,D,1}(k) - e_{i,D,2}(k)]^2}. \quad (29)$$

By the comparison of the condition with different values of N in the state of $N = 1$ and $N > 1$, we can find that the result after normalization is different. It is resulted from the different method of normalization. The former uses power estimation to normalize, while the latter uses the square root of power estimation to normalize.

3.3. *Stability Analysis of NCPAP-SSAF.* The stability of NCPAP-SSAF by analyzing the convergence of the algorithm will be presented in this subsection. We carry out the Taylor series expansion for $e_{i,D}(k+1)$ and get the following results according to the following equation [31]:

$$e_{i,D}(k+1) = e_{i,D}(k) + \frac{\partial e_{i,D}(k)}{\partial \alpha(k)} \Delta \alpha(k) + o(k), \quad (30)$$

where $o(k)$ stands for the higher order infinitesimal of the Taylor series. By rewriting the first-order quantities of Taylor expansion, it becomes as follows:

$$\frac{\partial e_{i,D}(k)}{\partial \alpha(k)} = \frac{\partial e_{i,D}(k)}{\partial \lambda(k)} \frac{\partial \lambda(k)}{\partial \alpha(k)}. \quad (31)$$

Substituting equations (12) and (13) into the updated equation (31), we get the following:

$$\frac{\partial e_{i,D}(k)}{\partial \lambda(k)} = y_{i,D,2}(k) - y_{i,D,1}(k). \quad (32)$$

By sorting out the preceding equation (16), we can find out the following relations:

$$\frac{\partial \lambda(k)}{\partial \alpha(k)} = \lambda(k)[1 - \lambda(k)]. \quad (33)$$

Substituting equations (32) and (33) into the update equation (31), we get the following:

$$\frac{\partial e_{i,D}(k)}{\partial \alpha(k)} = \lambda(k)[1 - \lambda(k)] [y_{i,D,2}(k) - y_{i,D,1}(k)]. \quad (34)$$

And $\alpha(k)$ can be calculated from equation (21):

$$\Delta \alpha(k) = -\gamma v_{t-1} - \mu_\alpha \lambda(k)[1 - \lambda(k)] \cdot \sum_{i=0}^N (y_{i,D,1}(k) - y_{i,D,2}(k)) \operatorname{sgn}[e_{i,D}(k)]. \quad (35)$$

From equations (34) and (35), we can get equation (36) when the subband number N is assumed to 1:

$$\begin{aligned} e_{i,D}(k+1) &= e_{i,D}(k) - \gamma v_{t-1} \lambda(k)[1 - \lambda(k)] \\ &\quad \cdot [y_{i,D,2}(k) - y_{i,D,1}(k)] + \mu_\alpha \lambda^2(k)[1 - \lambda(k)]^2 \\ &\quad \cdot (y_{i,D,2}(k) - y_{i,D,1}(k))^2. \end{aligned} \quad (36)$$

The result of the ideal filter should be that when k tends to ∞ , $e_{i,D}(k)$ tends to be 0. Then we can rewrite the expression equation (36) as follows:

$$\begin{aligned} |e_{i,D}(k+1)| &\leq |e_{i,D}(k) - \gamma v_{t-1} \lambda(k)[1 - \lambda(k)] \\ &\quad \cdot [y_{i,D,2}(k) - y_{i,D,1}(k)] + \mu_\alpha \lambda^2(k)[1 - \lambda(k)]^2 \\ &\quad \cdot (y_{i,D,2}(k) - y_{i,D,1}(k))^2|. \end{aligned} \quad (37)$$

Therefore, it can get the following equation:

$$\begin{aligned} \mu_\alpha \lambda^2(k)[1 - \lambda(k)]^2 (y_{i,D,2}(k) - y_{i,D,1}(k))^2 \\ \leq \gamma v_{t-1} \lambda(k)[1 - \lambda(k)] [y_{i,D,2}(k) y_{i,D,1}(k)]. \end{aligned} \quad (38)$$

So we conclude that when the mixing parameter satisfies the following conditions, the NCPAP-SSAF algorithm will converge according to the following equation:

$$0 \leq \mu_\alpha \leq \frac{\gamma v_{t-1}}{\lambda(k)[1 - \lambda(k)] [y_{i,D,2}(k) - y_{i,D,1}(k)]}. \quad (39)$$

3.4. *Computational Complexity and Memory Requirement.* To further illustrate the rationality of NCPAP-SSAF, it is necessary to test the algorithmic complexity and memory requirements of the algorithm. Concerning the multiplications, the algorithmic complexity is summarized in Table 1. All of these subband adaptive filtering algorithms require $3NL$ multiplication, and P represents the order of the analysis filter (synthesis filter). Most of algorithms' computation costs have been summarized in [16], so we mainly analyze CAP-SSAF and NCPAP-SSAF. Since both CAP-SSAF and NCPAP-SSAF require two filters so that tap-weight update requires $2MP + 4M/N$ multiplications and the subband error calculation requires $MP + 6M/N$ multiplications, for both CAP-SSAF and NCPAP-SSAF, the tap-weight vector can be rewritten as $\mathbf{w}(k) = \lambda(k)[\hat{\mathbf{w}}_1(k) - \hat{\mathbf{w}}_2(k)] - \hat{\mathbf{w}}_2(k)$ and thus they both require M/N multiplications. For each weight vector update, CAP-SSAF requires $2MP + 4M/N + (M+5)/N + 3NL + 1$ multiplications, and NCPAP-SSAF requires $2MP + 6M/N + (M+6)/N + 3NL + 1$ multiplication.

Besides, in Table 2, we analyze and compare the memory requirement of various algorithms. The NCPAP-SSAF combines two complete AP-SSAF filters and 8 independent parameters which are $[\alpha(k), \lambda(k), \mu_\alpha, \sigma(k), v, \eta]$. For $M = 512$, $L = 64$, $N = 4$, $P = 4$, it needs $2M(NP+1) + N(3L+6P+1) + 12 = 18288$ words to save the parameters of AP-SSAF and other NAG parameters.

4. Simulation Experiments and Results

4.1. *Setting Up the Environment of Experiment.* In this section, we will conduct two kinds of experiments. Firstly, the experiment of parameter analysis will show us the influence of key parameters on the performance of the algorithm and whether their performance is consistent with the theoretical expectation. Then we simulate the echo cancellation experiment to compare the performance of our proposed algorithm and other methods. The results of the experiments will verify that the proposed algorithm brings an improvement in both accuracy and convergence speed.

The sparse echo path used in the following experiments is shown in Figure 3, and both the sparse echo path and the adaptive filter have 512 coefficients with the sampling rate 8 kHz. In realistic communication scenarios, the impulse

TABLE 1: Algorithmic complexity.

| Algorithms | Multiplications | Computation |
|------------|------------------------------------|-------------|
| SSAF | $M + 2M/N + 3NL$ | 1536 |
| VSS-SSAF | $2M + (2M + 4)/N + 3NL$ | 2049 |
| ICNSAF | $4M + 4N + (M + 5)/N + 3NL + 3$ | 2964 |
| PAP-SSAF | $MP + 3M/N + 3NL$ | 3200 |
| CAP-SSAF | $MP + 4M/N + (M + 5)/N + 3NL + 1$ | 5506 |
| NCPAP-SSAF | $2MP + 6M/N + (M + 6)/N + 3NL + 1$ | 5762 |

$M = 512, L = 64, N = 4, P = 4.$

TABLE 2: Algorithmic memory requirements.

| Algorithms | Multiplications | Memory |
|------------|------------------------------------|--------|
| SSAF | $M(N + 1) + N(3L + 5) + 2$ | 3350 |
| ICNSAF | $2M(N + 1) + N(3L + 8) + 4$ | 5924 |
| CAP-SSAF | $2M(NP + 1) + N(3L + 6P + 1) + 8$ | 18284 |
| NCPAP-SSAF | $2M(NP + 1) + N(3L + 6P + 1) + 16$ | 18288 |

Memory unit: words

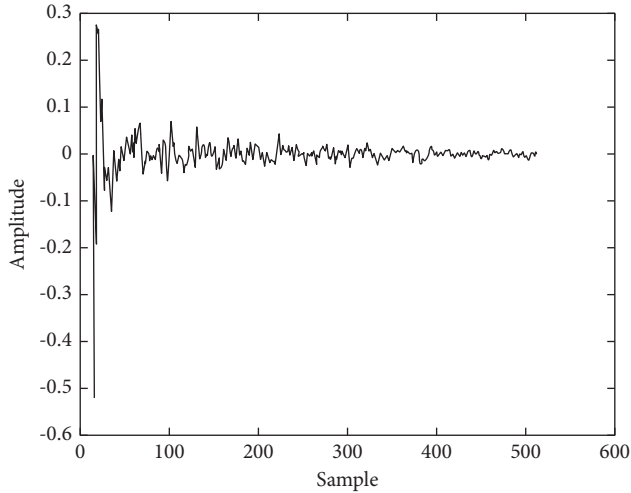


FIGURE 3: Echo impulse path in simulation. The x -axis represents the sampling point, and the y -axis represents the amplitude.

response of the echo path will be affected by environmental factors. Hence, we simulated an unexpected echo path impulse response by shifting the echo path to the right by 12 samples in the midst of each experiment, which is half of the total iteration number. Therefore, in the following experimental, we can see that the algorithms restart the convergence at the half process. The formula for array sparsity can be described as follows [32]:

$$\zeta = \frac{\sqrt{n} - \|x\|_1 / \|x\|_2}{\sqrt{n} - 1}. \quad (40)$$

By substituting the echo path vector and vector length into the formula, we find that the sparsity ζ is 0.6078.

There are two types of input signals: speech segments or an AR(1) process. The AR(1) signal is obtained by filtering a zero mean white Gaussian random sequence through the first-order system $H(z) = 1/(1 - 0.9z - 1)$, with the signal

length 6e4 points. Meanwhile, we use an independent white Gaussian noise with 30 dB signal-to-noise ratio and a strong impulsive noise with -10 dB signal-to-interference ratio as the system background noise and system output noise, respectively. The Bernoulli-Gauss distribution model is used to obtain impulse noise. The impulsive noise is generated as $z(k) = \omega(k)n(k)$, where $n(k)$ is Gaussian white noise with a mean value of 0 and a variance of δ , and $\omega(k)$ is a kind of Bernoulli process with occurrence probability $P\{\omega(k) = 1\} = P_r$, $P\{\omega(k) = 0\} = 1 - P_r$.

Double-talk is very common in echo cancellation. In order to simulate this scene, an 8 kHz sampling rate speech signal is added to near-end speaking in simulation. Figure 4 shows the signals of double-talk scenarios. Figure 4(a) is the near-end speech and Figure 4(b) is the far-end speech in all of the following experiments. In order to ensure a fair comparison, the following parameters were uniformly set in all algorithms, namely, subband $N = 4$, affine projection number $L = 4$, forgetting factor $\eta = 0.99$.

We did 50 times independent MonteCarlo in each simulation. We obtained the final results by averaging all of the 50 simulation results. The normalized mean square deviation (NMSD, in dB) was utilized to evaluate the convergence performance of the adaptive filters. It is defined as follows: $NMSD = 20\log_{10} E(\mathbf{w}(k) - \hat{\mathbf{w}}(k)_2) / \mathbf{w}(k)_2$

4.2. Momentum Parameter Analysis. Figures 5 and 6 show that the convergence curves which represent the NCPAP-SSAF with different γ in SNR = 20 dB, $P_r = 0.001$ for AR(1) input.

In Figure 5, we can see that different values of momentum factor γ cause different evolution results of mixing parameter $\lambda(k)$. The x -axis represents the number of iterations of the algorithm in the experiment, and the y -axis represents the value of NMSD. The larger momentum factor causes NAG to make a quicker choice between big steps and small steps. We can also see that $\lambda(k)$ is limited from 0.018 to 0.982 due to the fact that the absolute value of $\alpha(k)$ is less than 4. In Figure 6, PAP-SSAF and NCPAP-SSAF show much better performances than NLMS in terms of steady-state error and convergence speed. NAG is equivalent to the stochastic gradient method when $\gamma = 0$. It can be found from the figure that the “convergence pause” exits in the NCPAP-SSAF when $\gamma = 0$, which is marked in the circle. With an increment of the value of γ , the pause is reducing progressively to zero. Consequently, using NAG accelerates the convergence of the convex combination algorithm, and using $\gamma = 0.99$ gives a satisfying acceleration.

4.3. AR(1) Input. We carried out four different groups of experiments with AR(1) signal used as the input signal. In the following experiments, the μ_α of NCPAP-SSAF is set to 1.

Figure 7 shows the convergence of several algorithms under SNR = 20 dB and $P_r = 0.001$ conditions. The value of μ_α of ICNSAF and CAPSAF is set to 10. As can be seen from the figure, the convergence performance of PAP-SSAF is better than IP-SSAF with the same step size, which is undoubtedly due to the application of the affine projection. At the same

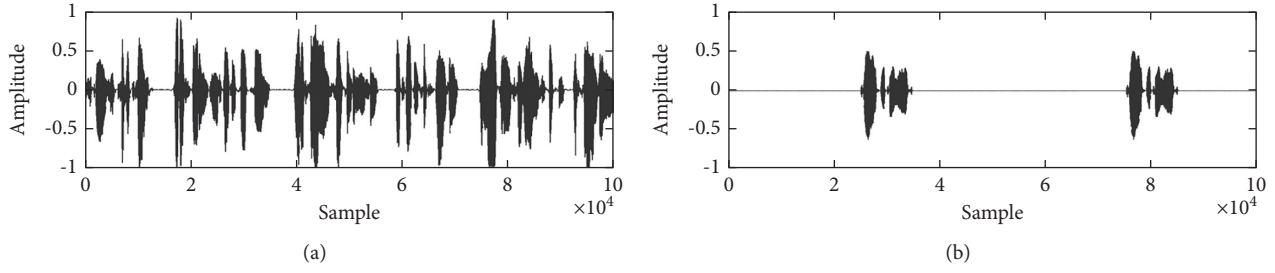


FIGURE 4: Representation in dual channel communication. The x -axis represents the sampling point, and the y -axis represents the amplitude of the speech signal. (a) Near-end speech. (b) Far-end speech.

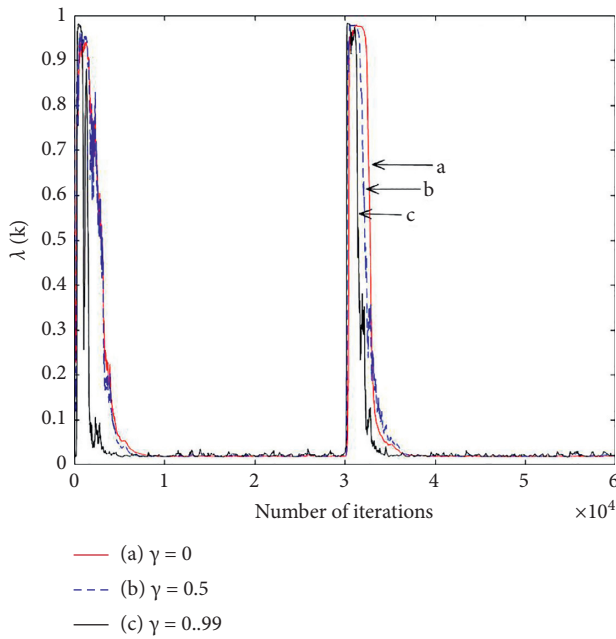


FIGURE 5: The evolution of mixing parameter $\lambda(k)$ with different γ . The x -axis represents the number of iterations of the algorithm in the experiment, and the y -axis represents the value of $\lambda(k)$.

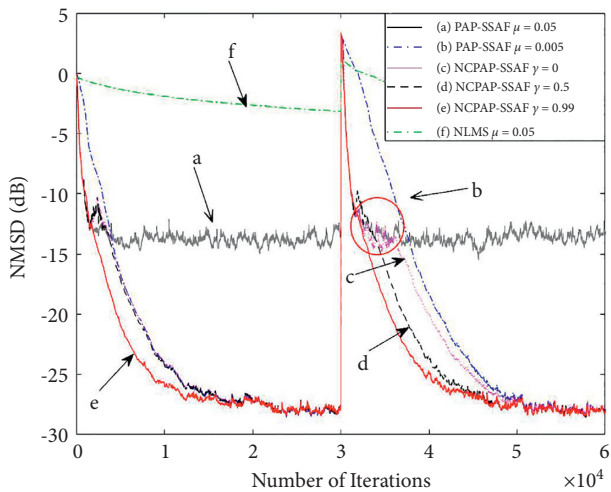


FIGURE 6: The convergence curves of NCPAP-SSAF with different γ . The x -axis represents the number of iterations of the algorithm in the experiment, and the y -axis represents the value of NMSD.

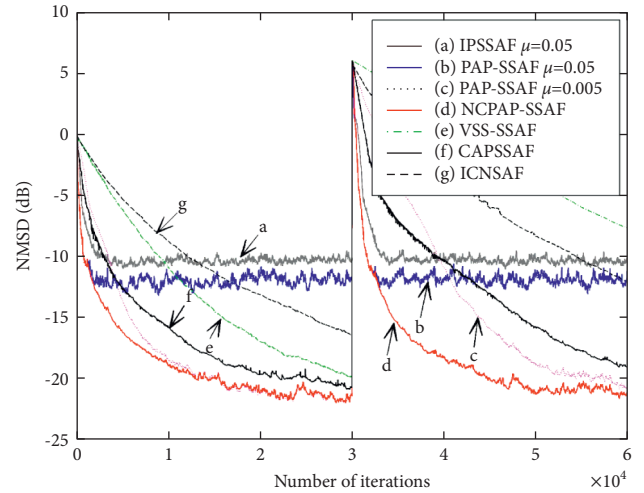


FIGURE 7: NMSD curves of various algorithms with $\text{SNR} = 20$ dB, $P_r = 0.001$ for AR(1) input.

time, the poor convergence performance of VSS-SSAF shows that in the sparse echo impulse response channel, a proportional step matrix is very necessary. Otherwise, the algorithm will not converge effectively. Compared with the algorithms without combinations, for the convex combinatorial algorithms, NCPAP-SSAF, CAP-SSAF, and ICN-SSAF, the performance of each has been improved, albeit to varying degrees. They have both fast convergence and low steady-state error and achieve the goal of algorithm design. By contrastively analyzing these three convex combination algorithms, it can be seen that the performance of NCPAP-SSAF proposed in this paper is significantly better than other algorithms. Its convergence rate is the same as that of large step PAP-SSAF, and its steady-state error is the same as that of small step PAP-SSAF. In addition, due to the application of the NAG method, NCPAP-SSAF does not appear the “pause-convergence” phenomenon in the process of convergence. NCPAP-SSAF has the normalized mixing step-size convex combination structure, which contains two PAP-SSAF with different step sizes. Its excellent performances should not only be attributed to the own structure but also the characteristic of PAP-SSAF, namely, a faster convergence speed.

Figure 8 shows the convergence performance of several algorithms in $\text{SNR} = 20$ dB and $P_r = 0.01$. The values of μ_α at

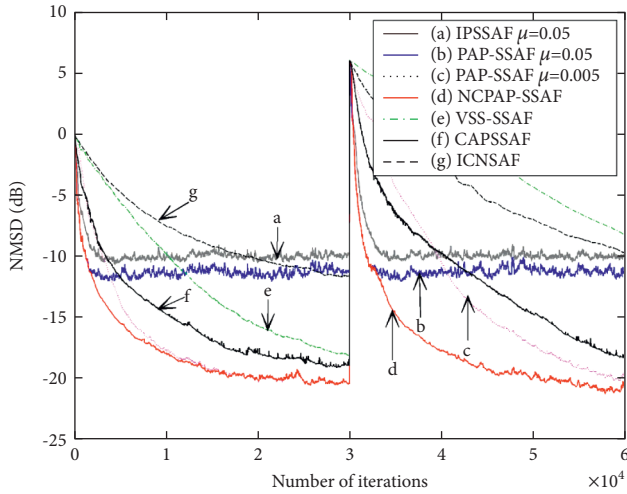


FIGURE 8: NMSD curves of various algorithms with SNR = 20 dB, $P_r = 0.01$ for AR(1) input.

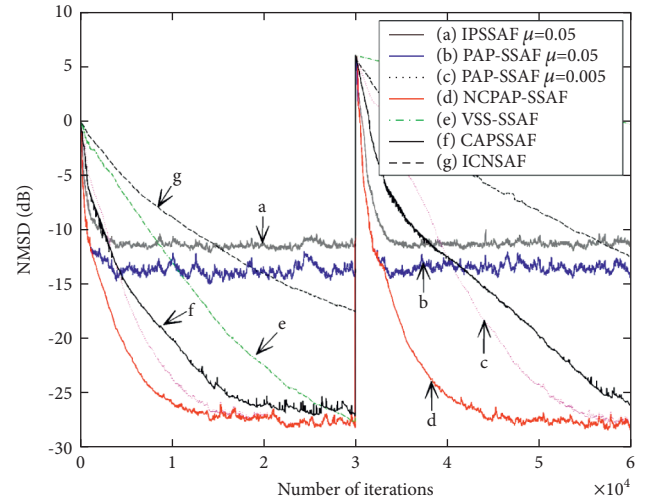


FIGURE 9: NMSD curves of various algorithms with SNR = 30 dB, $P_r = 0.001$ for AR(1) input.

ICNSAF and CAPSAF are set to 100. The performance of these algorithms is approximately the same as that of Figure 7. NCPAP-SSAF performs well in resisting impulse noise, and its performance is obviously better than other algorithms. Please note that in Figures 7 and 8, the steps of mixing parameters used by NCPAP-SSAF are the same, and all of them have achieved good performance. That means that the normalized step size proposed in this paper makes the combined filter not affected by impulse noise. Hence, the algorithm has good robustness. With the P_r increasing, the steady-state errors of other algorithms increase in varying degrees.

Figures 9 and 10 show the convergence performance of AR(1) input of these algorithms under the low noise conditions. In Figure 9, the SNR = 30 dB, $P_r = 0.001$, and in Figure 10, the SNR = 30 dB, $P_r = 0.01$. By comparing Figures 7 and 8, we can see that the steady-state errors of these algorithms decrease in varying degrees with the increase of SNR. NCPAP-SSAF achieves good results in two experiments with the same step size of mixing parameter. Compared with Figures 9 and 10, it can be seen that with the increase of the probability of impulse noise, the convergence speed and steady-state error of several algorithms decrease, but the algorithms still maintain a good ability of anti-impulse noise. Comparing the experimental results of Figures 9 and 10 with those of the previous two experiments, it can be seen that with the increase of SNR, all the algorithms achieve a better performance.

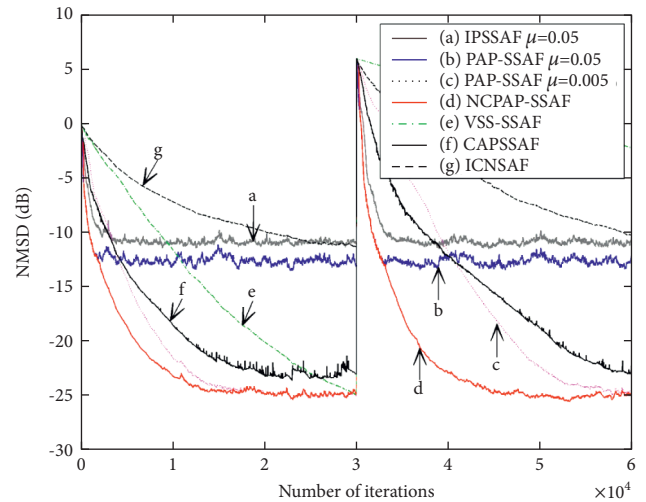


FIGURE 10: NMSD curves of various algorithms with SNR = 30 dB, $P_r = 0.01$ for AR(1) input.

4.4. Speech Input. Taking a voice signal as input, four independent experiments were carried out in total. Figures 11 and 12 show the simulations with the single-talk case, and Figures 13 and 14 show the double-talk case because the correlation of speech signals is much greater than that of white noise, which passes through the first-order systems.

The value of μ_α at NCPAP-SSAF in these experiments is set at 0.05. The performance of the NCPAP algorithm is discussed and analyzed in the following.

Figures 11 and 12 show under different SNRs the convergence of several algorithms without near-end voice. The values of μ_α at ICNSAF and CAP-SSAF are set at 100 in Figure 11, and the values of μ_α at ICNSAF and CAP-SSAF are set at 5000 in Figure 12.

As shown in both figures, we can conclude the conclusion that the NCPAP-SSAF algorithm has much better performance in the aspects of fast convergence speed and small steady-state error. Compared with the condition when $\mu = 0.005$ at PAP-SSAF, it is clear that when $\mu = 0.005$, it gradually converges to a certain extent (about 12 dB) and remains stable when $\mu = 0.05$. This is because the larger step size makes it impossible for the adaptive filter to continue to converge. For the condition that $\mu = 0.005$, its former process has too slow convergence speed and too poor tracking performance, although it can obtain a smaller steady-state error.

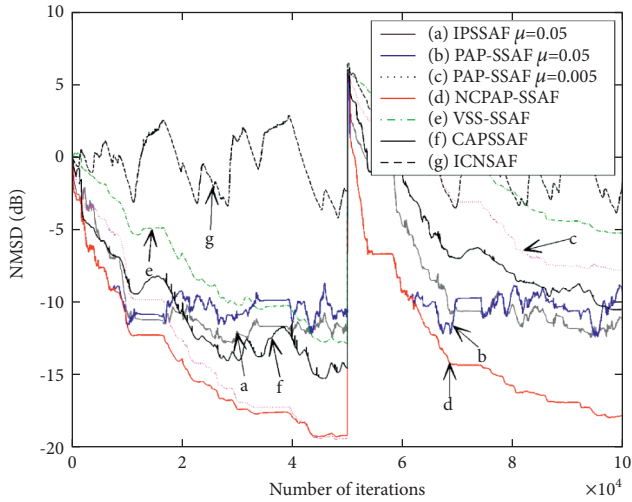


FIGURE 11: NMSD curves of various algorithms with SNR = 20 dB, $P_r = 0.001$ for single-talk speech input.

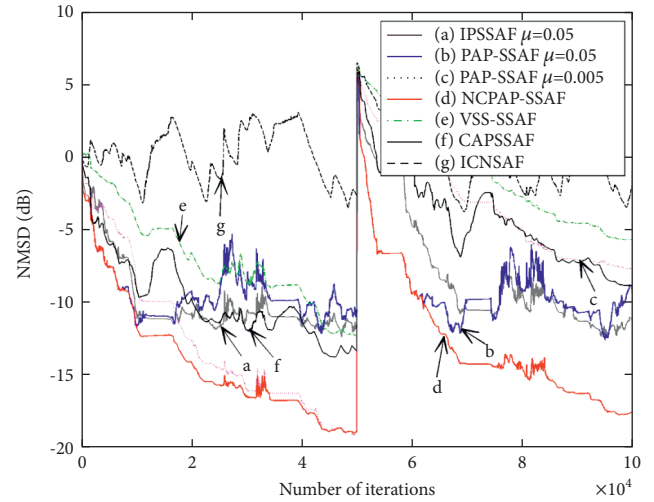


FIGURE 13: NMSD curves of various algorithms with SNR = 20 dB, $P_r = 0.001$ for double-talk speech input.

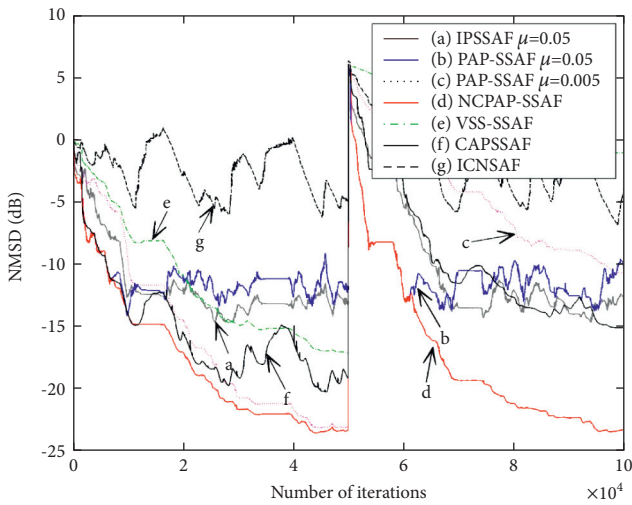


FIGURE 12: NMSD curves of various algorithms with SNR = 30 dB, $P_r = 0.001$ for single-talk speech input.

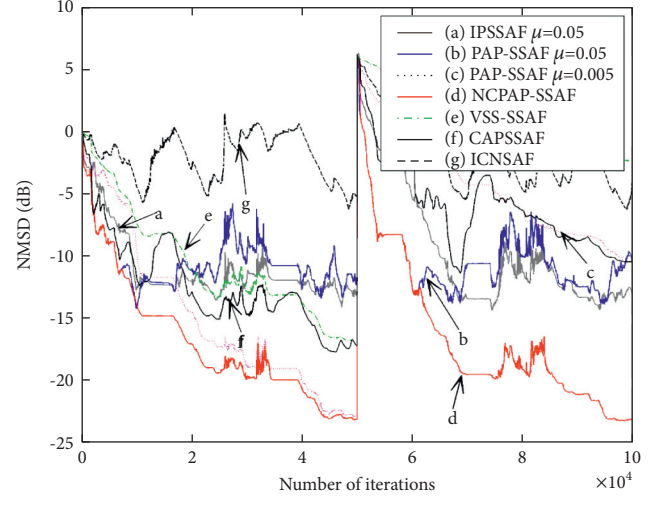


FIGURE 14: NMSD curves of various algorithms with SNR = 30 dB, $P_r = 0.001$ for double-talk speech input.

The situation indicates that the NCPAP-SSAF algorithm adaptively matches a digital filter with large step size and can constantly converge when the filter is close to the steady state and also can acquire similar steady-state error with the PAP-SSAF when the value of μ is 0.05. In addition, in the two experiments, the ICNSAF algorithm and the CAP-SSAF algorithm need to choose a different value of μ_α in order to get the best filtering effect. For its filter with normalized mixing parameter step size, the NCPAP-SSAF algorithm can get the best filtering effect without adjusting the value of μ_α and improve the robustness of the system and reduce the influence of external factors. In the case of single-talk voice input, the NCPAP-SSAF algorithm can combine two independent filters scientifically and reasonably and has good robustness.

Figures 13 and 14 show the convergence of several algorithms with a near-end voice under different SNR, respectively. The values of μ_α at ICNSAF and CAP-SSAF are

set at 100 in Figure 13, and the values of μ_α at ICNSAF and CAP-SSAF are set at 5000 in Figure 14.

It can be seen from both figures that NCPAP-SSAF has obvious performance advantages, fast convergence speed, and small steady-state error, and it is not disturbed by near-end voice.

First, for PAP-SSAF with $\mu = 0.05$ and PAP-SSAF with $\mu = 0.005$, it can be clearly seen that the former has a large steady-state error and is disturbed by the near-end speech, which results in the divergence of the filter in a certain degree. And though the latter can get a smaller steady-state error, the convergence speed is too slow especially when the echo path changes. Then as for NCPAP-SSAF, it can be found that the convergence curve of NCPAP-SSAF in the initial convergence stage of the filter almost coincides with that of PAP-SSAF with $\mu = 0.05$. It indicates that NCPAP-SSAF adaptively chooses a filter with a larger step size, which is consistent with the design goal. Next, NCPAP-SSAF can

keep fast convergence speed when the filter reaches steady-state gradually, and the filter can remain stable when the near-end voice appears, which indicates that NCPAP-SSAF has a certain antijamming ability. Being consistent with the previous results, NCPAP-SSAF achieves good filtering performance and improves the robustness of the system by using the same mixing parameter step size in both experiments. Compared with Figure 11 and Figure 12, it can be seen that with the increase of SNR, the performances of all adaptive filtering algorithms have been improved to a certain extent, especially in reducing the steady-state error. So we can reach the conclusion that the proposed algorithm has better double-talk robust than the other two combination filter methods, which owes the l_1 -norm as the cost function.

5. Discussion

The proposed method has some referenced effects on echo cancellation. With a convex combination of two independent filters, the NCPAP-SSAF algorithm exhibits a far superior filtering performance. However, there are still some issues that need to be further improved in the later work. One limitation of the current systems is that the computational complexity of the algorithm is similar to that of an algorithm based on combination filters but much larger than that of a single filter. This can be improved by upgrading composite structure, simplifying unnecessary calculation courses and reducing the complexity. Moreover, all the experiments in this paper are carried out in the MATLAB simulation environment. And impulse noise is generated by the Gaussian Bernoulli distribution based simulation, which is not consistent with the impulse noise in the real world.

6. Conclusion

In this paper, we design a new combination structure called NCPAP-SSAF for affine projection symbolic subband adaptive filtering algorithm. The power normalization method with the mixing parameter step size we proposed improves the robustness of the algorithm. We do simulation experiments of echo cancellation to validate the effectiveness of the proposed algorithm. First, we test the influence of the momentum factor on the mixing parameter. Then, we compare the performance of our proposed algorithm and other methods. The simulation results show that the NCPAP-SSAF algorithm is not affected by stationary noise or impulse noise to a certain extent, and it can accurately obtain the optimal combination parameters under different conditions and thus obtain the optimal filtering performance. In the case of double-talk speech, NCPAP-SSAF can maintain faster convergence speed and smaller steady-state error and has a certain ability to resist near-end speech interference and strong robustness. Compared with other algorithms, our proposed method accelerates the convergence speed, reduces the steady-state error, and improves the robustness. In future research, we will improve the combination structure and simplify unnecessary calculations to reduce the computational complexity of the algorithm.

Abbreviations

| | |
|-----------|--|
| LMS: | Least mean squares |
| NLMS: | Normalized least mean squares |
| APA: | Affine projection algorithm |
| NSAF: | Normalized subband adaptive filter |
| SSAF: | Sign subband adaptive filter |
| VRP-SSAF: | Variable regularization parameter SSAF |
| AP-SSAF: | Affine projection sign subband adaptive filter |
| IWF-SSAF: | Individual-weighting-factor SSAF |
| P-SSAF: | Proportionate SSAF |
| ICNSAF: | Improved convex combination normalized subband adaptive filter |
| PAP-SSAF: | Proportionate affine projection sign subband adaptive filter |
| SLMS: | Sign-error LMS |
| NAG: | Nesterov's accelerated gradient. |

Data Availability

The data were generated according to the method described in this paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61350009), and in part by the National Natural Science Foundation of China 61179045.

References

- [1] R. Vanamadi and A. Kar, "Feedback cancellation in digital hearing aids using convex combination of proportionate adaptive algorithms," *Applied Acoustics*, vol. 182, Article ID 108175, 2021.
- [2] M. T. Akhtar, F. Albu, and A. Nishihara, "Maximum Vectorial-criterion (MVC)-based adaptive filtering method for mitigating acoustic feedback in hearing-aid devices," *Applied Acoustics*, vol. 181, Article ID 108156, 2021.
- [3] S. H. Pauline, D. Samiappan, R. Kumar, A. Anand, and A. Kar, "Variable tap-length non-parametric variable step-size NLMS adaptive filtering algorithm for acoustic echo cancellation," *Applied Acoustics*, vol. 159, 2020.
- [4] C. Shi, N. Jiang, R. Xie, and H. Li, "A simulation investigation of modified FxLMS algorithms for feedforward active noise control," in *Proceedings of the 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1833–1837, Lanzhou, China, November 2019.
- [5] Z. Zheng and H. Zhao, "Affine projection m-estimate subband adaptive filters for robust adaptive filtering in impulsive noise," *Signal Processing*, vol. 120, pp. 64–70, 2016.
- [6] I. J. C. Eun Jong Lee, "A new sign subband adaptive filter with improved convergence rate," *The Journal of the Acoustical Society of Korea*, vol. 33, no. 5, pp. 335–340, 2014.
- [7] J. Ni, X. Chen, and J. Yang, "Two variants of the sign subband adaptive filter with improved convergence rate," *Signal Processing*, vol. 96, no. 5, pp. 325–331, 2014.

- [8] R. L. Das and V. Trivedi, "An adaptive upper threshold based gain function for the ZA-PNLMS algorithm," *IEEE Transactions on Circuit and System II-Express Brifs*, vol. 67, no. 10, pp. 2274–2278, 2020.
- [9] U. Mahbub, S. A. Fattah, W. P. Zhu, and M. O. Ahmad, "Single-channel acoustic echo cancellation in noise based on gradient-based adaptive filtering," *EURASIP Journal on Audio Speech and Music Processing*, vol. 2014, no. 1, p. 20, 2014.
- [10] J. Ni and F. Li, "Variable regularization parameter sign subband adaptive filter," *Electronics Letters*, vol. 46, no. 24, p. 1605, 2010.
- [11] P. Wen and J. Zhang, "Robust variable step-size sign subband adaptive filter algorithm against impulsive noise," *Signal Processing*, vol. 139, pp. 110–115, 2017.
- [12] Y. Yu and H. Zhao, "Novel sign subband adaptive filter algorithms with individual weighting factors," *Signal Processing*, vol. 122, pp. 14–23, 2016.
- [13] S. H. Kim, J. J. Jeong, J. H. Choi, and W. K. Sang, "Variable step-size affine projection sign algorithm using selective input vectors," *Signal Processing*, vol. 115, no. C, pp. 151–156, 2015.
- [14] Y. Yu and H. Zhao, "Memory proportionate APSA with individual activation factors for highly sparse system identification in impulsive noise environment," in *Proceedings of the Sixth International Conference on Wireless Communications and Signal Processing*, pp. 1–6, Hefei, China, October 2014.
- [15] Y. Yu, T. Yang, H. Y. Chen, R. C. de Lamare, and Y. S. Li, "Sparsity-aware SSAF algorithm with individual weighting factors: performance analysis and improvements in acoustic echo cancellation," *Signal Processing*, vol. 178, 2021.
- [16] F. Albu and H. K. Kwan, "Memory improved proportionate affine projection sign algorithm," *Electronics Letters*, vol. 48, no. 20, pp. 1279–1281, 2012.
- [17] M. Chandra, P. Goel, A. Anand, and A. Kar, "Design and analysis of improved high-speed adaptive filter architectures for ECG signal denoising," *Biomedical Signal Processing and Control*, vol. 63, 2021.
- [18] Y. Song, Y. Z. Ren, X. L. Liu, W. L. Gao, S. Tao, and L. Guo, "A nonparametric variable step-size subband adaptive filtering algorithm for acoustic echo cancellation," *International Journal of Agricultural and Biological Engineering*, vol. 13, no. 3, pp. 168–173, 2020.
- [19] S. Burra and A. Kar, "Performance analysis of an improved split functional link adaptive filtering algorithm for nonlinear AEC," *Applied Acoutics*, vol. 176, 2021.
- [20] J. Lu, Q. Zhang, W. Shi, and L. Zhang, "Variable step-size normalized subband adaptive filtering algorithm for self-interference cancellation," *Measurement Science and Technology*, vol. 32, no. 9, 2021.
- [21] J. W. Yoo, J. W. Shin, and P. G. Park, "Variable step-size affine projection sign algorithm," *IEEE Transactions on Circuits and Systems II Express Briefs*, vol. 61, no. 4, pp. 274–278, 2014.
- [22] L. Lu and H. Zhao, "Adaptive combination of affine projection sign subband adaptive filters for modeling of acoustic paths in impulsive noise environments," *International Journal of Speech Technology*, pp. 1–11, 2016.
- [23] R. Vanamadi and A. Kar, "Feedback cancellation in digital hearing aids using convex combination of proportionate adaptive algorithms," *Applied Acoutics*, vol. 182, Article ID 108175, 2021.
- [24] Z. Zheng, Z. Liu, H. Zhao, Y. Yu, and L. Lu, "Robust set-membership normalized subband adaptive filtering algorithms and their application to acoustic echo cancellation," *IEEE Transactions on Circuits and Systems I Regular Papers*, vol. 64, no. 99, pp. 1–14, 2017.
- [25] F. R. Yang, G. Enzner, and J. Yang, "New insights into convergence theory of constrained frequency-domain adaptive filters," *Circuits, Systems, and Signal Processing*, vol. 40, no. 4, pp. 2076–2090, 2020.
- [26] L. Xiao-meng, S. Gao-ping, and Q. Xiao-hui, "Improved subband adaptive filter and its application in echo cancellation," *Chinese Signal Processing*, vol. 32, no. 8, pp. 973–981, 2016.
- [27] S. Koike, "Adaptive step-size q-normalized least mean modulus-Newton algorithm," in *Proceedings of the 2016 IEEE Region 10 Conference (TENCON)*, pp. 1158–1161, Singapore, Asia, November 2016.
- [28] T. Zhang, H. Q. Jiao, and Z. C. Lei, "Individual-activation-factor memory proportionate affine projection algorithm with evolving regularization," *IEEE Access*, vol. 5, no. 99, pp. 4939–4946, 2017.
- [29] Y. Nesterov, "Implementable tensor methods in unconstrained convex optimization," *Mathematical Programming*, vol. 186, no. 1-2, pp. 157–183, 2021.
- [30] P. W. Wen and J. S. Zhang, "Variable step-size diffusion normalized sign-error algorithm," *Circuits, Systems, and Signal Processing*, vol. 37, no. 11, pp. 4993–5004, 2018.
- [31] L. Lu, H. Zhao, Z. He, and B. Chen, "A novel sign adaptation scheme for convex combination of two adaptive filters," *AEUE - International Journal of Electronics and Communications*, vol. 69, no. 11, pp. 1590–1598, 2015.
- [32] L. Yang and Z. Yang, "Incremental robust non-negative matrix factorization with sparseness constraints and its application," *Journal of Computer Applications*, vol. 39, no. 5, pp. 1275–1281, 2019.

Research Article

IOT-Based Cotton Whitefly Prediction Using Deep Learning

Rana Muhammad Saleem ¹, Rafaqat Kazmi ², Imran Sarwar Bajwa ², Amna Ashraf ²,
Shabana Ramzan ³ and Waheed Anwar ²

¹Department of Computer Science, University of Agriculture, Faisalabad Sub Campus, Burewala, Pakistan

²Department of Software Engineering, The Islamia University of Bahawalpur, Bahawalpur, Pakistan

³Department of Computer Science, The Government Sadiq College Women University, Bahawalpur, Pakistan

Correspondence should be addressed to Imran Sarwar Bajwa; imran.sarwar@iub.edu.pk

Received 20 July 2020; Revised 29 April 2021; Accepted 29 June 2021; Published 12 July 2021

Academic Editor: Javid Taheri

Copyright © 2021 Rana Muhammad Saleem et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Agriculture is suffering from the problem of low fertility and climate hazards such as increased pest attacks and diseases. Early prediction of pest attacks can be very helpful in improving productivity in agriculture. Insect pest (whitefly) attack has a high influence on cotton crop yield. Internet of Things solution is proposed to predict the whitefly attack to take prevention measures. An insect pest prediction system (IPPS) was developed with the help of the Internet of Things and a RBFN algorithm based on environmental parameters such as temperature, humidity, rainfall, and wind speed. Pest Warning and Quality Control of Pesticides proposed an economic threshold level for prediction of whitefly attack. The economic threshold level and RBFN algorithm are used to predict the whitefly attack using temperature, humidity, rainfall, and wind speed. The seven evaluation metrics accuracy, *f*-measures, precision, recall, Cohen's kappa, ROC AUC, and confusion matrix are used to determine the performance of the RBFN algorithm. The proposed insect pest prediction system is deployed in the high influenced region of pest that provides pest prediction information to the farmer to take control measures.

1. Introduction

The Food and Agricultural Organization (FAO) predicts that the world population will reach 8 billion people by 2025 and 9.6 billion people by 2050 [1]. Due to an increase in population, the need of foods is increasing day by day. The basic needs (foods etc.) of humans cannot be met by using old traditional farming methods. The old farming methods consume more manpower and are less efficient. The risk of less productivity is still there by using old traditional farming methods. The crop yield can be increased by using new farming methods with the usage of IoT technology. The "Internet of Things" (IoT) is a creative idea integration through which any object can transfer data through the network [2,3]. The "Internet of Things" (IoT) is an exceptionally encouraging group of innovations that are capable of offering numerous solutions towards the modernization of agriculture [4]. Agriculture is one of the sectors that is

expected to be highly influenced by the advances in the domain of IoT.

Cotton is one of the most well-known fiber crops. Cotton is a significant business crop around the world. It is known as the lord of fiber and is designated "as white gold" assuming a key job in numerous socio-economic parts of the world. Cotton is cultivated in 77 countries of the world including Pakistan [5]. Cotton crop employs millions of farmers and workers. It supports the cotton textile industry of Pakistan. It provides not only fiber for the textile industry but also cotton seed which is a major source of edible oil for human consumption. Its proteinaceous oil cake is used as a food supplement for dairy animals.

The cotton crop yield is influenced by many kinds of insect pests. Whitefly (*Bemisia tabaci*) is one of them, having a severe effect on cotton crop yield [6]. The attack of whitefly is a global problem for cotton crop with extreme danger to worldwide sustainable agricultural development [7].

Sustainable agriculture is a new farming concept which is based on scientific innovations to fulfill the need of food and textile needs of society. Climatic conditions have a potential impact on whitefly [8]. IoT application is suitable to increase the cotton crop yield with the monitoring of environmental conditions [3,9]. IoT can perform a significant role in the prediction of whitefly based on environmental factors.

As the agricultural sector is facing many challenges regarding climate change, the current challenges of the less favorable climatic conditions flourish the more serious hazards for cotton crops. The climatic conditions influence crop production which results in the economic loss to the farmers [10]. Therefore, continuous monitoring of the climatic conditions is suggested to reduce the attack of whitefly on cotton crop. Different environmental parameters such as temperature, humidity, rainfall, and wind speed could be monitored through IoT technology. As a result, whitefly prediction would be held through the data of environmental parameters and deep learning algorithms.

Figure 1 shows the flow of the pest attack prediction process at any crop. In the first step, it shows the types of crops such as cotton, wheat, rice, and sugarcane, which are the main crops of Pakistan. In Figure 1, step 2 represents the stages of crop growth such as seed planting, germination, sprout, plant, flowers, and fruit. In Figure 1, 3rd step illustrates the environmental parameters such as temperature, humidity, rainfall, and wind speed which have effects on different stages of crop growth. The 4th step shows the growth of pests after the climatic effect on different stages of crop growth.

- (i) Whitefly's attack time/month is an important factor because in a specific time/month, the growth of whitefly is probably more
- (ii) Environmental factors such as (temperature, humidity, rainfall, and wind speed) are major factors for the growth of whitefly insect pests [11]
- (iii) Whitefly insect pest growth is more in a dry environment
- (iv) High wind speed can transfer the larva and eggs from one place/area to another

The study aims is to seek protection from insect pest attacks with the early prediction using environmental factors to produce high crop yield and to alert the farmer of taking prevention measures. The remaining paper is structured into "related work," "methodology," "implementation," and "results and discussion."

2. Related Work

During the literature, we will discuss the smart agriculture system especially focusing on different insect pest predictions and their solutions.

Raghavendra et al. [12] focused on weather-based prediction of pest in cotton by using different prediction model machine learning algorithms such as multiple linear regression (REG) and generalized linear model (GLM). The authors also applied statistical correlation on weather

parameters. The weather parameters were consisting of maximum temperature, minimum temperature, morning humidity, evening humidity, and rainfall. The experiments were conducted on both training and test datasets. The weather parameters' data used were from 2006 to 2010.

Shang et al. [13] proposed a prediction model to predict the occurrence of insect pests by combining the two machine learning algorithms: artificial neural network and genetic algorithm. The prediction model included three parts i.e., input layer, hidden layer, and output layer. The author used meteorological data (precipitation, sunshine hours, mean temperature, relative humidity, and so on) as an input to predict the occurrence of insect pests. The author claimed 91.67% accuracy of prediction of insect pests due to an intelligent model and by applying a hybrid algorithm.

Kim et al. [14] presented a FaaS (Farm as a Service) model which consists of EMS (Equipment Management Service), DMS (Data Management Service), MMS (Model Management Service), FMS (Smart Farm Monitoring Service), FCS (Smart Farm Control Service), and FOS (Smart Farm Operation Service). This model had been used to predict pest and disease in the strawberry crop, in which image capture devices and sensors were used for the prediction of pests and disease in crops. The authors analyzed the environmental factors, pathogens, and host plants and simulated the captured data. The captured data was processed and analyzed through the FaaS Model. This model predicted the pest and disease occurrence and sent the alarm to the farmer through the mobile-based service.

Sajjad et al. [15] focused on the effect of climate change (temperature, humidity, rainfall, and sunshine) on major crops (e.g., sugarcane, maize, rice, and wheat) in Pakistan. The authors stated that the high temperature has a severe negative impact and low temperature has a positive impact on major crop production. The standard error techniques HAC (heteroskedasticity and autocorrelation) and FGLS (feasible generalized least square) were used to find the regression result.

Tripathy et al. [16] predicted the pest/disease, collected the sensory data which consists of temperature, humidity, leaf wetness, and soil moisture, and continuously monitored these parameters. The authors have developed a decision support system by using DM (data mining technique) and multivariate regression mining algorithm for groundnut crop. The authors developed a prediction model and these above techniques were used in this model. The authors claimed to achieve high accuracy in the prediction of pest attack.

Rubanga et al. [17] focused on small-scale greenhouse farming. Due to the shortage of labor in Japan, there is a dire need to develop a smart agricultural system. The smart system gathered real-time climate data through a wireless sensor network (WSN) and stored it in a web-based database. The stored data is used to calculate, analyze, and formulate the whole data which displayed the result on screen. The result is displayed in the form of a graph for the ease of understanding to help the farmer in decision-making to increase the production of tomatoes. The growing degree day (GDD) algorithm was used to calculate and analyze the microclimate environment in the greenhouse.

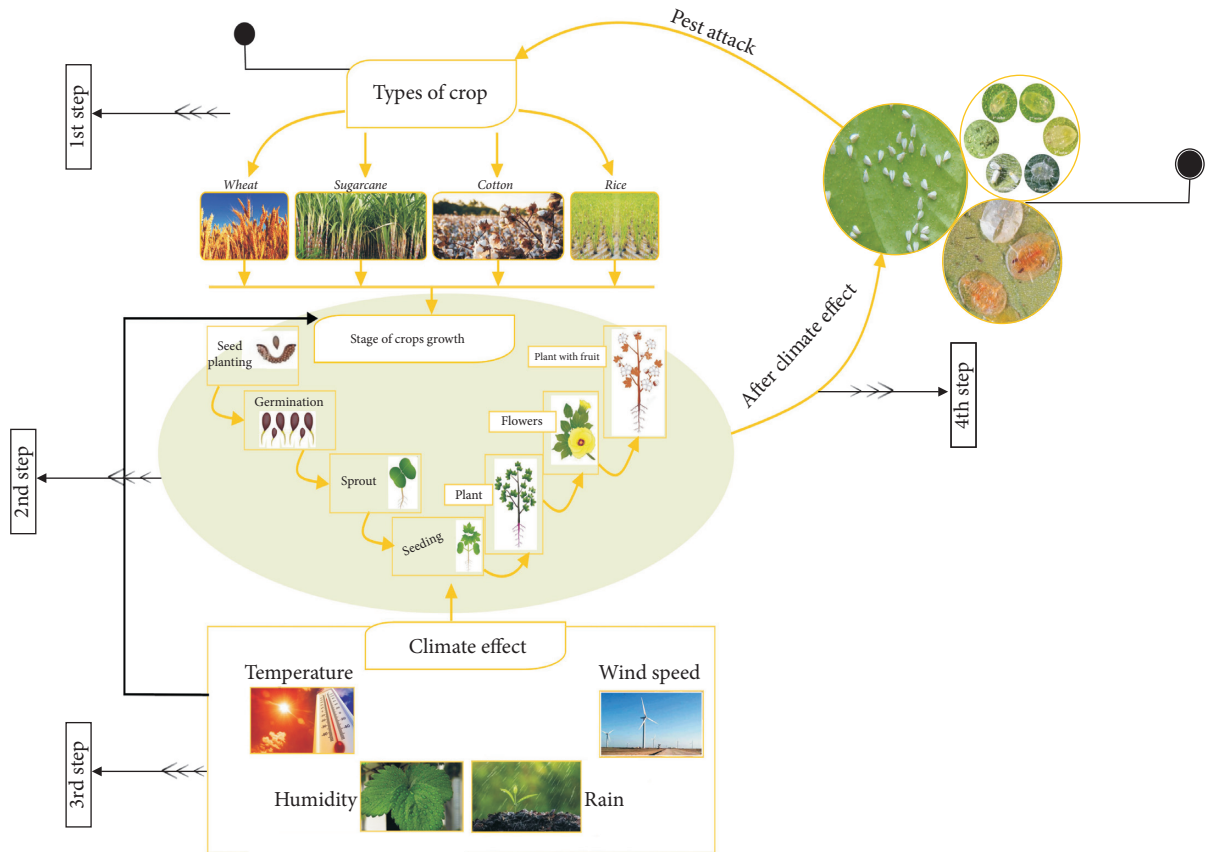


FIGURE 1: Climate role in growth of whitefly insect pests.

Wang et al. [18] developed a dynamic simulation model for the prediction of rice brown plant hoppers by using historical metrological data. The model consisted of complex features that were dynamic, persistent, nonlinear, and multivariable. There were five inputs (temperature, humidity, rainfall, intensity of light, and sunshine hours) to the simulation model, and one output result was the growth rate of brown plant hoppers. The authors claimed to achieve the high accuracy for the prediction of rice brown plant hoppers.

Mekala et al. [19] presented different techniques to boost up the agriculture market by using CLAY-MIST measurement techniques which were based on the sensed temperature and humidity to assess the comfort level of the crop. It presents the IoT cloud model which shows 5-layer architecture. The results are gathered by using different hardware such as micro-controller, sensor, communication protocols, and IoT cloud servers, and the authors developed a CMM algorithm for measurement of the CLAY-MIST index. This algorithm found the issues, calculated accurate decisions about issues, and sent a report to the farmers. The outcomes were 94% exact with less execution time when compared with the current warm comfort strategies.

Trogo et al. [20] presented agriculture as a major industry in every country. The use of technology had great influence to increase the yield. They used a smart agricultural solution, called DSSAT. DSSAT used Automated Weather Station (AWS) sensors and SMS technology, as well as expert knowledge of the farmers. The use of technologies such as SMS played a vital role to alert to the farmers. With the usage of technology, climate alerts, dry soil, and fertilizer alerts could be made accessible to the farmers.

Mathurkar et al. [21] presented that the agricultural sector performs a crucial role in the economics of every country. These days everything operates automatically. There are sensors to operate farms automatically. The enhancement in the crop yield could be done with the usage of FPGA. By using sensor devices, sense the data such as moisture level, temperature, and humidity and apply FPGA to monitor the environmental and soil condition required to know the timings of water supply to the fields for better growth of plants.

Table 1 presents the detail of previous studies which display the techniques/sensors and purpose of the study about the pest monitoring and prediction system.

TABLE 1: Relevant prediction methods in the previous study.

| Study | Year | Sensor/method used | Objective |
|--------------------------|------|--|---|
| Li et al. [22] | 2008 | No sensors/ISODATA iterative self-organizing data analyzed technique algorithm | Prediction of disease/insect pests for Guangdong vegetables |
| Wei and Lin [23] | 2009 | No sensor/fuzzy radial basis function neural network | Pest predicting |
| Li et al. [24] | 2010 | No sensor/maximum likelihood algorithm | Forecast model for vegetable pests |
| Raghavendra [12] | 2014 | No sensor/multiple linear regression and generalized linear model | Prediction of pests in cotton |
| Lee et al. [25] | 2017 | No sensor/correlation between pests and weather | Prediction for multiple crops |
| Li et al. [26] | 2020 | Image processing | Crop pest recognition in natural scenes using convolutional neural networks |
| Liu and Wang [27] | 2020 | Image processing | Tomato diseases and pests detection based on improved Yolo V3 convolutional neural network |
| Xiao et al. [28] | 2019 | No real time/use weather dataset | Occurrence prediction of pests and diseases in cotton on the basis of weather factors by long short term memory network |
| Türkoğlu and Hanbay [29] | 2019 | No sensor/image processing | Plant disease and pest detection using deep neural network |
| He et al. [30] | 2019 | Camera and light source/imaging system | Detect oilseed rape pests based on deep learning |

- (i) Previous studies focused more on insect pest detection not prediction
- (ii) Previous studies focused more on image processing-based pest prediction
- (iii) Previous studies more focused on insect pest disease identification or detection
- (iv) Previous studies had not focused on the accuracy of prediction
- (v) Previous studies had not used sensors for insect pest prediction
- (vi) Intelligent decision-making was not used for prediction

3. Methodology

This portion explained about the model and design of the suggested solution about whitefly insect pest prediction, methods, and algorithm.

3.1. Architecture of Suggested Prediction Method. The suggested insect pest prediction method is designed with the capability of intelligent categorization of environmental factors such as temperature, humidity, rainfall, and wind speed. Our approach focuses on efficient energy consumption as it does not turn on all the sensors all the time. The proposed approach precisely monitors the whitefly insect pest growth environment by classifying the environmental factors by continuing with a RBFN. The proposed architecture consists of several layers, as shown in Figure 2. It describes that the five layers of the prediction system are input layer, gateway layer, storage layer, prediction layer, and application layer. The implementation of each layer is given below.

3.1.1. Input Layer. In Figure 2 hierarchical structure, the first layer consists of sensors of different environmental parameters such as temperature, humidity, rain, and wind

speed. The data is collected by using sensors of these abovementioned parameters. These sensors are deployed at an experimental plot.

3.1.2. Gateway Layer. In Figure 2 hierarchical structure, the second gateway layer consists of a different hardware device such as a microcontroller (Arduino). The microcontroller is responsible for collecting data measured by sensors. The Wi-Fi module is responsible for the transfer of the data to the IoT server.

3.1.3. Storage Layer. In Figure 2 hierarchical structure, the third storage layer consists of an IoT server. The previous gateway layer transfers the data to the IoT server and stores the data in MySQL. The MySQL data can be exported in the CSV form.

3.1.4. Prediction Layer. In Figure 2 hierarchical structure, the fourth prediction layer consists of a machine learning algorithm on the IoT server for the prediction of whitefly insect pest by using exported data from MySQL in the CSV form. At this layer, RBFN, a deep learning algorithm is deployed for prediction.

3.1.5. Application Layer. In Figure 2 hierarchical structure, the fifth application layer consists of predicted output which is displayed or transferred to the Android application for farmers to take necessary action.

In Figure 2, all layers have a strong relationship. The first input layer is basically related to the perception layer in IOT architecture in which it consists of sensors. This input layer can send data through the gateway layer using Wi-Fi modules to the server and store data at the storage layer. The next prediction layer is basically a processing layer in which the RBFN algorithm develops and makes

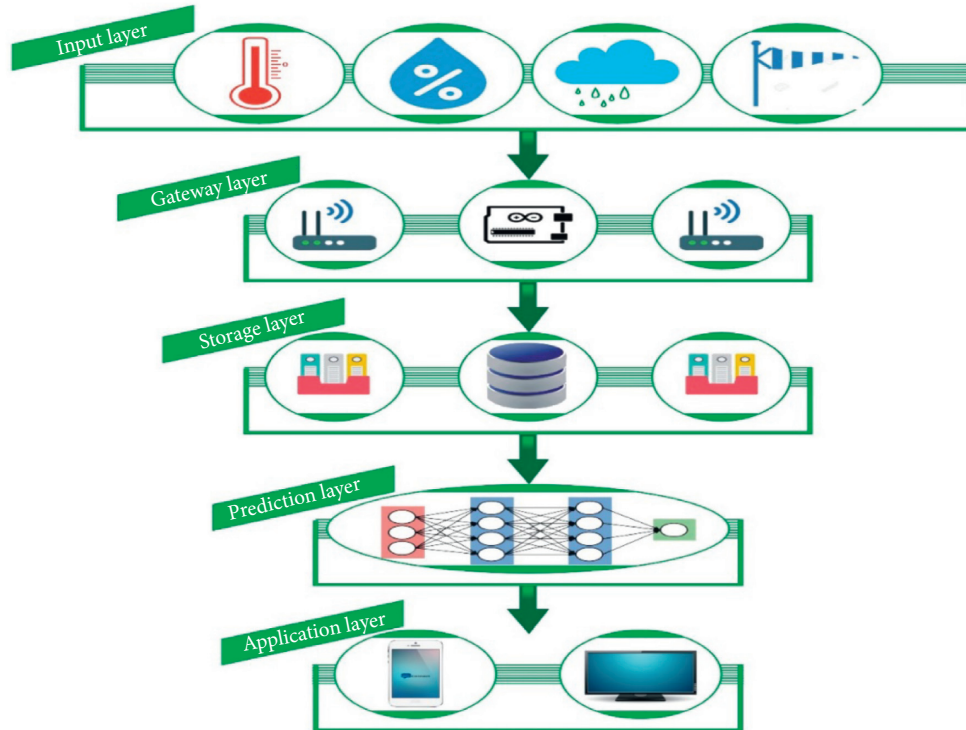


FIGURE 2: Layered architecture of the proposed model.

prediction using test and train data which are obtained from the storage layer, and the last layer is the application layer in which prediction about pest can be sent to the farmers.

The strength point of the problem is that the sensors sensed real-time data from the field and addressed the unique idea to predict the pest attack in the field. Farmers can take prevention measures according to the environment.

There are many key technologies of IoT but the main key technologies used in architecture are RFID at the gateway layer, sensor technology at the input layer, communication technology Wi-Fi modules at the gateway layer, and cloud computing at the storage layer.

3.2. Radial Basis Function Network for Prediction of Whitefly Insect Pest. RBFN is used as a deep learning algorithm for getting the proposed prediction. RBFN is selected due to its performance in binary class prediction with the independence condition of datasets. Deep learning is too beyond the machine learning. The machine learning algorithm already has the solution of binary classification but the suggested model uses RBFN due to five reasons.

- (i) RBFN has supremacy in terms of accuracy when trained with a huge amount of data. Keeping in view with the passage of time, the data of environmental parameters such as temperature, humidity, rainfall, and wind speed have been increased.
- (ii) More reliable when a huge amount of data are processed through RBFN.

- (iii) Accuracy will be increased when the amount of data size has been increased.
- (iv) RBFN techniques have an efficient decision support system for prediction.
- (v) RBFN has high accuracy with a complex problem.

RBFN consists of many layers. Details are given below:

- (i) The first layer is the input layer where environmental parameters are given
- (ii) The second layer is a hidden layer that consists of one layer where processing or learning has been performed
- (iii) The last layer is the output layer where output/prediction is displayed

The regression or classification problem could be resolved with the use of the RBFN Algorithm. The most significant benefit of deep learning is that feature extraction has been performed automatically. Deep learning has a great influence on the industry and agricultural sectors.

Any neuron in the neural network has two parts, as shown in Figure 3. One is the calculated linear function, and the other is to calculate the activation function.

The linear function in weights' nonlinear function is known as activation function.

Figure 4 describes the structure of RBFN which has one input layer, one hidden layer, and one output layer.

Compute linear and nonlinear/activation function of the hidden layer:

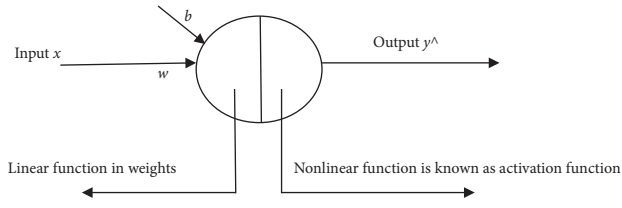


FIGURE 3: Neuron in RBFN.

$$f(x) = \sum_{j=1}^m w_j h_j(x), \quad (1)$$

$$h(x) = \exp\left(-\left(\frac{(x-c)^2}{r^2}\right)\right). \quad (2)$$

In (1) and (2), $f(x)$ is the output which consists of x as inputs and w_j as the weight of the hidden layer multiplied by Gaussian activation function $h(x)$, $h(x)$ is a Gaussian activation function with the parameter r radius of the neuron, and c is the center defined separately at each RBF unit.

RBFN consists of three layers i.e., one input layer, one hidden layer, and one output layer.

- (i) Input layer is called X , which consists of input data from sensors taken by Pest Environment Monitoring System (PEMS). The input data consists of temperature, humidity, rainfall, and wind speed.
- (ii) Output layer is displays the output in the form of Yes or No.

4. Implementation

This section shows experimental settings, layout of experimental area, prototype model deployment in an experimental area, and RBFN implementation.

4.1. Experimental Area. The hardware is deployed in the city of Faisalabad of Pakistan. The location of Pakistan in the world is presented in Figure 5.

4.2. Crop, Season, and Insect Pest. The scientific name of cotton is *Gossypium hirsutum*. Pakistan is a cotton-growing land. Due to the attack of whitefly pest, the cotton growth is decreasing with the passage of time. The suggested solution is deployed for the prediction of the whitefly pest [31]. The suggested solution is extendable to any other insect pest. The selected area has two cropping seasons. To conduct the experiment, 2nd season is selected which prevails from May to November.

4.3. Experimental Plot Layout. The selected area for the experiment is one acre (43,560 ft²) having length and width of 208 × 208 feet. To observe the whitefly attacks, a dataset having 416 rows and 62 columns was used. Each column has 416 plants. One foot space has 2 cotton plants. Total cotton plants are 12,896 for the prediction of the whitefly

population. The layout detail of the experimental area is shown in Figure 6.

4.4. Equipments Used. Temperature, humidity, rainfall, and wind speed sensors are used for the model to execute the suggested solution. The attributes of the sensor devices describe the predictive features.

4.4.1. Temperature and Humidity Sensor. Figure 7 presents a temperature sensor device known as DHT-22 to produce highly accurate data (sense the temperature and humidity) from the atmosphere. DHT-22 device is low cost and low powered. DHT-22 provides numeral results. The technical details of DHT-22 with characteristics are mentioned in Table 2.

4.4.2. Rain Detection Sensor Device. In our suggested model, the usage of the rainwater detection sensor device is displayed in Figure 8 which is a low cost, low powered, and lightweight device for measuring the intensity of rainwater in the open air. The rainwater detector sensor provides both digital and analog output. The rain sensor module is an easy tool for rain detection. The technical details of the rainwater detector with characteristics are mentioned in Table 3.

4.4.3. Anemometer/Wind Speed Sensor Device. In our suggested model, the usage of the wind speed sensor device is displayed in Figure 9, which is a reliable and stable sensor for measuring the intensity of wind speed in the open air. The anemometer/wind speed sensor provides an analog output. The wind speed sensor is an easy tool for wind speed measurement. The technical details of the wind speed sensor with characteristics are mentioned in Table 4.

4.4.4. Microcontroller. In our suggested model, a microcontroller having name WeMos D1 Wi-Fi UNO-based ESP8266 shield for Arduino, as shown in Figure 10, with characteristics in Table 5 is used.

4.5. Prototype Model and Deployment. The prediction model is shown in Figure 11, with Wi-Fi Arduino and sensors. The developed model is used to observe the whitefly attack with the effect of environmental factors such as temperature, humidity, rainfall, and wind speed values. The hardware model is developed and deployed in the crop field, as shown in Figure 11.

The web application is developed using the PHP Language; MySQL is implemented at the IoT webserver. The web application captures the environmental data, processes it, and stores it. Four times in a day, data of sensors are captured through sensors. The libraries “ESP8266WiFi.h” and “DHT.h” are used in a web application to send data from the sensors to the server (May to November) and store data in the database, as shown in Figure 12.

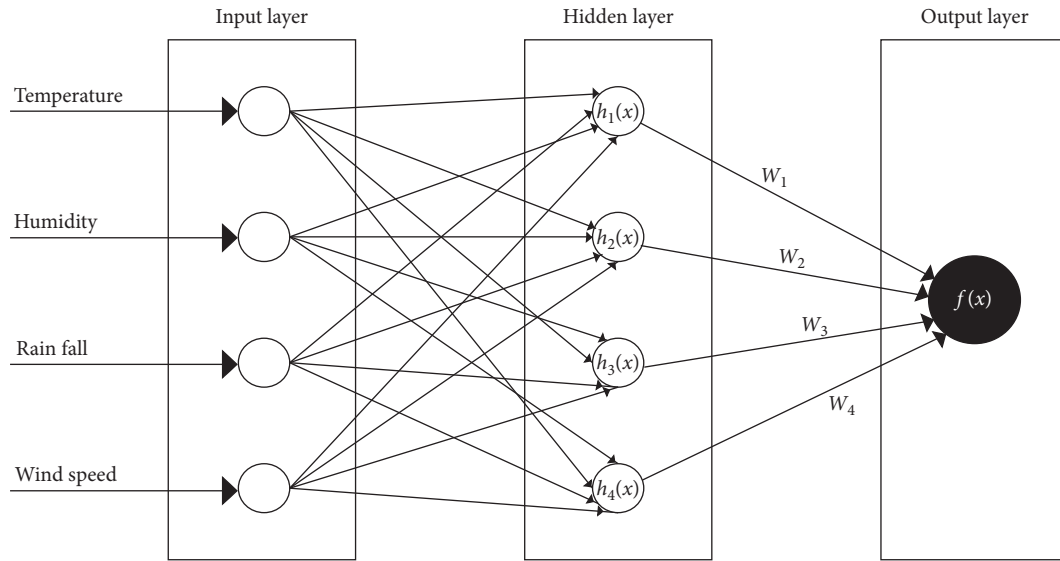


FIGURE 4: RBFN.



FIGURE 5: Pakistan’s location in the world map.

4.6. RBFN Implementation. Optimum values for cotton crop of environmental factors such as temperature, humidity, rainfall, and wind speed with respect to insect pest whitefly are given in Table 6.

Whitefly has positive relation with temperature and wind speed, while negative relation with humidity and rainfall. The whitefly population increases with the increase of temperature and wind speed, while the whitefly population decreases with the increase of relative humidity and rainfall [11].

4.7. Display Result. The predicted output has been displayed at the Android app for farmers to take further necessary action to control the whitefly pest at the initial level. The sample output message is shown in Figure 13.

5. Results and Discussion

Our suggested model (RBFN) has the capability of decision-making to predict the attacks of the whitefly. The deployment of layered design and prototype is depicted in the

earlier sections. The four sensors (temperature, humidity, rainfall, and wind speed) and a microcontroller have been deployed on the selected zone to assess the result. Cotton is developed in May and finishes in November in the selected zone. The deployment time of hardware is from May to November of 2018 and from May to November of 2019. The proposed model captured the temperature data as displayed in Figures 14 and 15 for the year 2018 and 2019 with daily maximum temperature, every day minimum temperature, and daily average temperature.

Figures 14 and 15 plot the maximum, minimum, and average temperatures from 1st May to 1st December for the year of 2018 and 2019. Maximum temperature was 47°C and 48°C in June month and minimum temperature was 5°C and 4°C in November for the year of 2018 and 2019, respectively. The plotted graph represents the maximum temperature in the blue line, the minimum temperature in the red line, and the average temperature in the green line. The X-axis represents the time interval and Y-axis represents temperature in Celsius.

Figure 16 shows daily data about average temperature, humidity, rainfall, and wind speed from May to Nov of 2018 captured from sensors, stored in the database, and then downloaded in the CSV form.

Figure 17 shows daily data about average temperature, humidity, rainfall, and wind speed from May to Nov of 2019 captured from sensors, stored in the database, and then downloaded in the CSV form.

5.1. Performance of RBFN Model. Performance of the RBFN is detected in terms of accuracy, precision, recall, and *f*-measure. Precision is the fraction of the correct prediction out of the total prediction made, and recall is the ratio of the accurate prediction to all the prediction in the binary class. The execution of RBFN is performed in Python language by using the “Keras” library. The achievement of the RBFN algorithm is measured by using the “sklearn.metrics” library

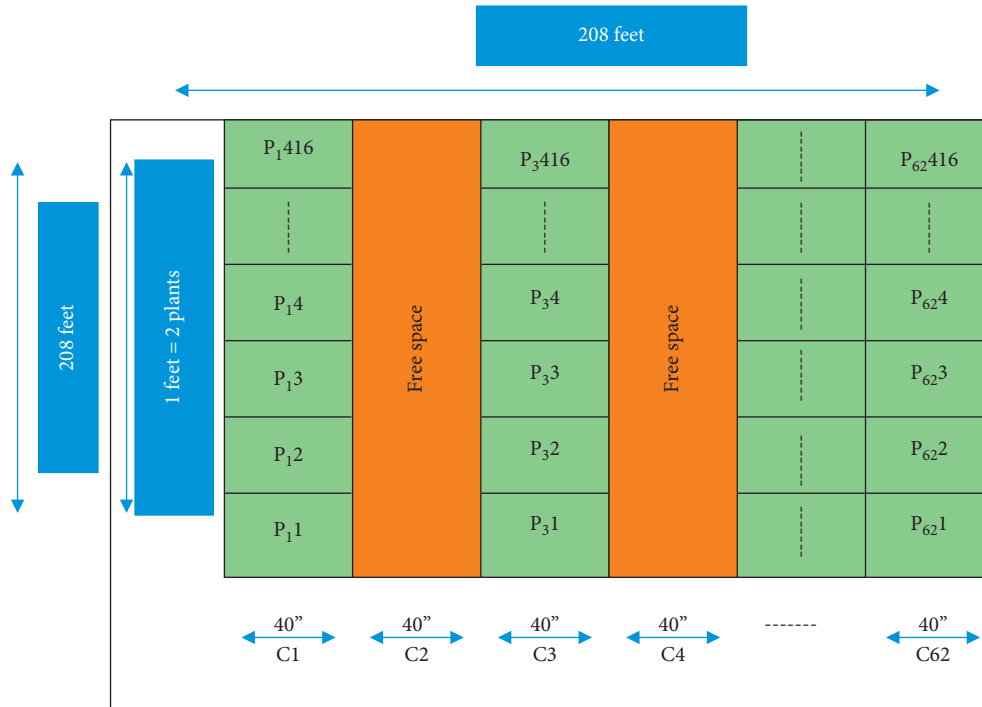


FIGURE 6: Sampling point for insect pest whitefly observation in the field.

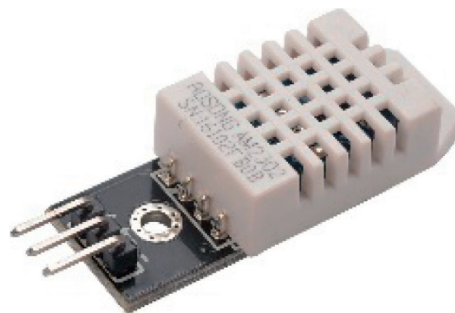


FIGURE 7: DHT22 temperature and humidity sensor device.

in Python with the accuracy of RBFN 82.88%, high F_1 , precision, recall, support, Cohen’s kappa, ROC AUC, log loss, and confusion matrix of different predictive features, as shown in Table 7 and 8, respectively.

The actual and predicted values of 2018 and 2019 plotted, as shown in Figure 18.

Figure 18 plots the output as expected and predicted results. The plotted graph represents the expected (actual) results in the blue line and the predicted results in the orange line. It is observed that there is a fluctuation in the results which is why we have not got 100% accuracy in the result. The X-axis represents the record of test data, and the Y-axis represents the prediction output from 0 to 1.

5.2. Field Evaluation of Proposed Model. Field evaluation is performed by observing the economic threshold level (ETL) of whitefly of 5 adult or nymph or both per leaf [32] in the fields at different times. One-acre sampling points of the

cropping area plotted from May to November of 2018 and 2019 in the experiment plot and observation were started at the start of May. Infield evaluation below the ETL mean does not exist and above the ETL mean exists. The maximum whitefly has been observed in July 15, 2018 to August 15, 2018 and May 15, 2019 to August 30, 2019. In the selected experimental area, a maximum of 24 whiteflies per leaf has been observed. These 24 whiteflies consist of different forms of whitefly such as egg, pupal, and adults. The observed whitefly intensity is presented in Table 9.

The intensity of whitefly in May to Nov. 2018 has been observed during field evaluation, as shown in Figure 19.

During the field evaluation, the population of whitefly has been observed, and the observed data are plotted in Figure 19. In Figure 19, the maximum intensity of whitefly has been observed during July and August in the year of 2018. The plotted graph represents the X-axis as a time interval and the Y-axis as the intensity of the whitefly population.

TABLE 2: Characteristics of the DHT-22 sensor device.

| DHT-22 temperature and humidity sensor device | |
|---|-----------------|
| Length and width | 1.5 cm × 2.5 cm |
| Voltage | 3 V to 5 V |
| Maximum current flow | 2.5 mA |
| Temp. measuring Range | -40~80°C |
| Humidity | 0~100% |
| Temperature measurement precision | ±0.5°C |
| Humidity measurement precision | ±2% RH |

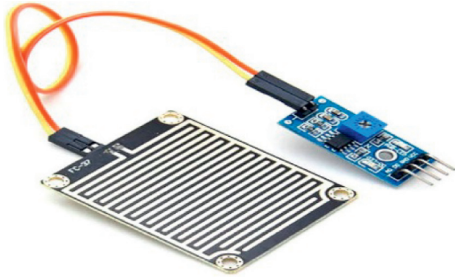


FIGURE 8: Rainwater detector sensor.

TABLE 3: Characteristics of the rainwater detector sensor.

| Rainwater detector sensor device | |
|----------------------------------|--|
| Driver dimensions | 32 mm × 15 mm × 9 mm ($L \times W \times H$) |
| Collector board size | 54 mm × 40 mm × 1.5 mm ($L \times W \times H$) |
| Power | 3.3–5 V |



FIGURE 9: Wind speed measure sensor.

TABLE 4: Characteristics of the wind speed sensor.

| Wind speed sensor | |
|--------------------------------------|----------------|
| Max current | 4–20 mA/0–5 V |
| Power supply | DC12–24 V |
| Start wind speed | 0.2 m/s |
| Effective wind speed measuring range | 0 to 30 m/s |
| Sensor styles | Three cups |
| Signal output way: pulse current | 4–20 mA/0–5 V |
| Working temperature | -40 C~80 C |
| Transmission distance | More than 1 km |

The intensity of whitefly in May to Nov. 2019 has been observed during field evaluation, as shown in Figure 20.

Again in 2019, population of whitefly has been observed during field evaluation, and the observed data are plotted in Figure 20. In Figure 20, the maximum intensity of whitefly has been observed during the end of July and start of the

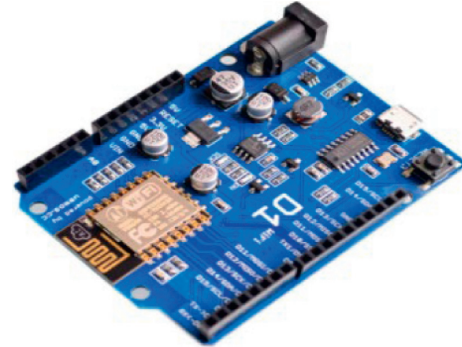


FIGURE 10: Microcontroller (Arduino).

TABLE 5: Characteristics of a microcontroller (Arduino).

| Microcontroller (Arduino) | |
|---------------------------|-------------------|
| Microcontroller | ESP-8266EX |
| Operating voltage | 3.3 V |
| Clock speed | 80 MHz/160 MHz |
| Dimension | 68.6 mm × 53.4 mm |
| Weight | 25 g |
| Digital I/O pins | 11 |
| Analog input pins | 1 |

August in the year of 2019. The plotted graph represents the X-axis as a time interval and the Y-axis as the intensity of the whitefly population.

Hot spots of whitefly above ETL have been observed during a field evaluation of May to Nov. 2018 in the experimental area, as shown in Figure 21.

Figure 21 displays different whitefly hotspot points in the experimental area. It shows different whitefly intensity levels for different experimental points. The graph plots in three-dimension represent the length, width of the plot, and intensity of the whitefly population. The 72.5% hotspots of whitefly above ETL have been observed during field evaluation in the experimental area in 2018.

Hot spots of whitefly above ETL have been observed during a field evaluation of May to Nov. 2019 in the experimental area, as shown in Figure 22.

Figure 22 also displays different whitefly hotspot points in the experimental area for the year of 2019. It shows different whitefly intensity levels for different experimental points. The above graph plots in three-dimension represent the length, width of the plot, and intensity of the whitefly population. The 74.5% hotspots of whitefly above ETL have been observed during field evaluation in the experimental area in 2019.

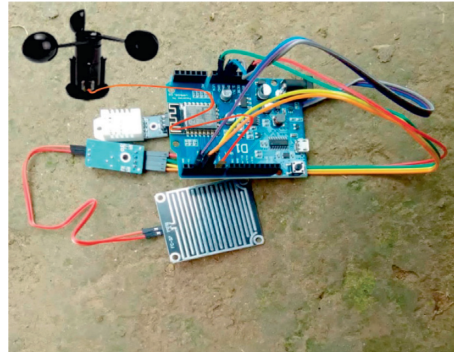


FIGURE 11: Hardware model.

| id | dated | tmax | tmin | humidity | rain | wind_speed |
|----|---------------------|------|------|----------|------|------------|
| 1 | 2018-05-01 09:00:00 | 28.5 | 28.5 | 15.5 | 0 | 2.7 |
| 2 | 2018-05-01 15:00:00 | 39 | 28.5 | 19.5 | 0 | 2.8 |
| 3 | 2018-05-01 21:00:00 | 22.5 | 22.5 | 20 | 0 | 3 |
| 4 | 2018-05-02 03:00:00 | 21.5 | 21.5 | 15 | 2.8 | 5 |
| 5 | 2018-05-02 09:00:00 | 31 | 21.5 | 17 | 0 | 4 |
| 6 | 2018-05-02 15:00:00 | 36.5 | 21.5 | 19 | 0 | 3.6 |
| 7 | 2018-05-02 21:00:00 | 32 | 21.5 | 17 | 0 | 4.1 |
| 8 | 2018-05-03 03:00:00 | 34.5 | 34.5 | 27 | 0 | 3.9 |
| 9 | 2018-05-03 09:00:00 | 39.5 | 34.5 | 30 | 0 | 2.6 |
| 10 | 2018-05-03 15:00:00 | 22.5 | 22.5 | 18 | 4 | 4.3 |
| 11 | 2018-05-03 21:00:00 | 25 | 22.5 | 23 | 0 | 3.2 |

FIGURE 12: Sensor data stored on the server database.

TABLE 6: Optimum values of environmental factors.

| Abiotic factors | Optimum values |
|----------------------|----------------|
| Temperature [11, 32] | 35–51°C |
| Humidity [11, 32] | Below 65% |
| Rainfall [11] | 1–2 mm |
| Wind speed [11] | 5.50–5.75 km/h |

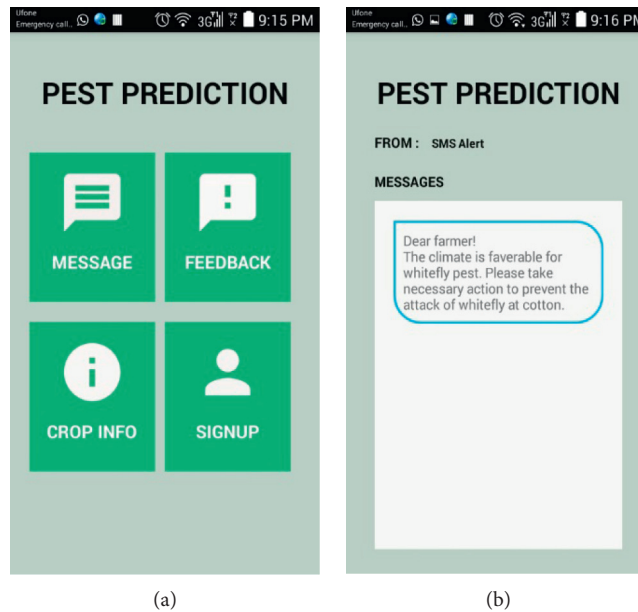


FIGURE 13: Android application for displaying predication message to the farmer.

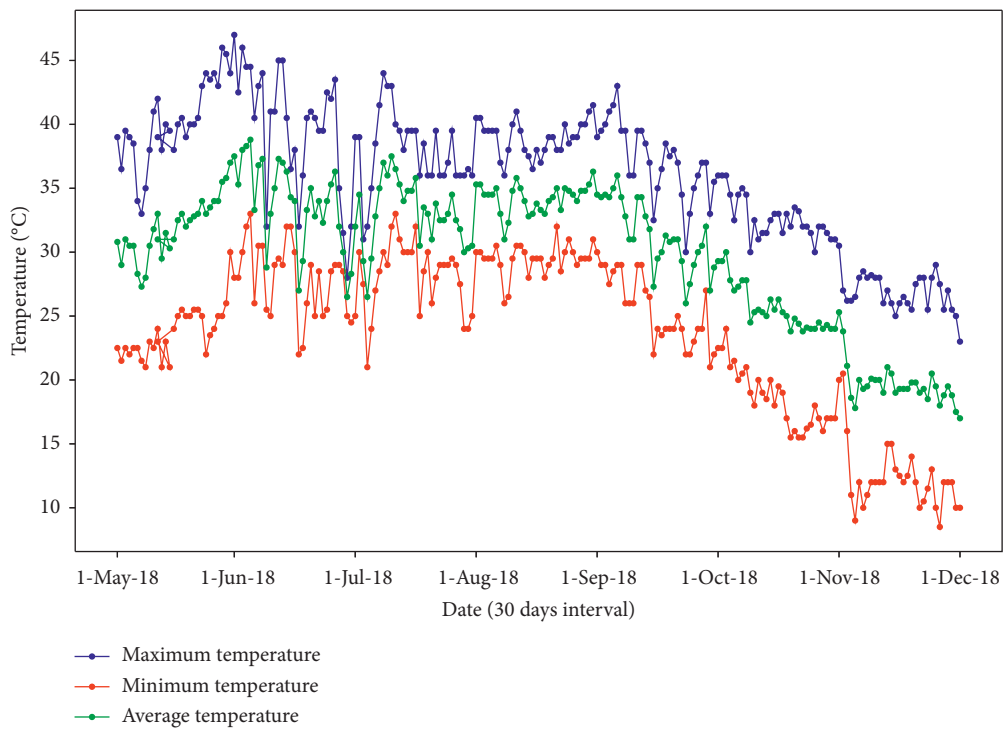


FIGURE 14: Maximum, minimum, and average temperature from May to Nov. 2018.

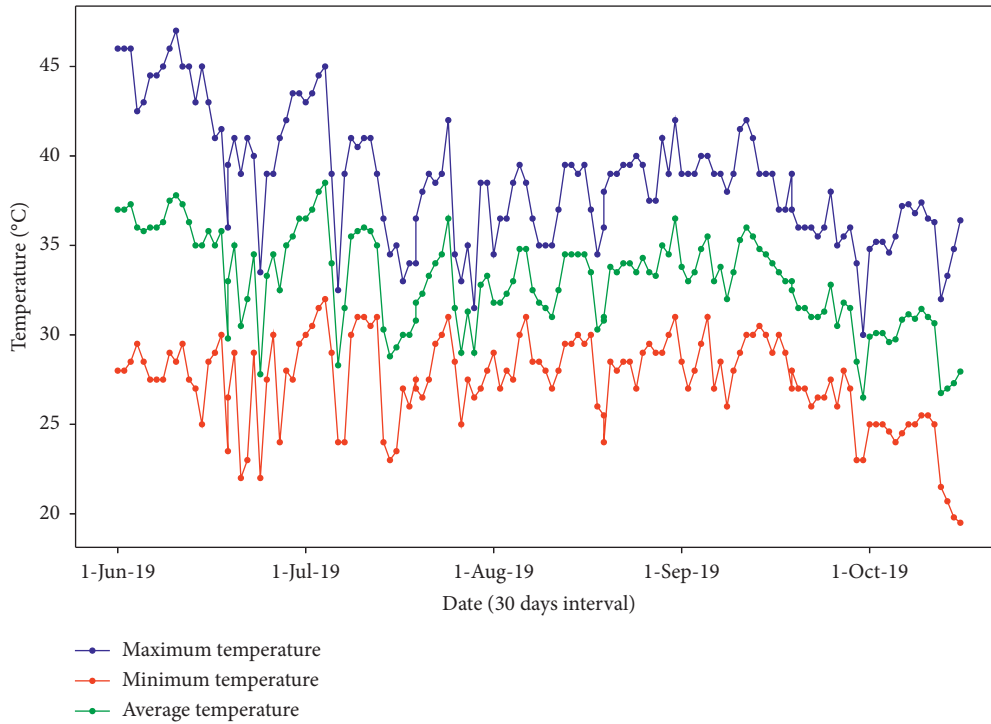


FIGURE 15: Maximum, minimum, and average temperature from May to Nov. 2019.

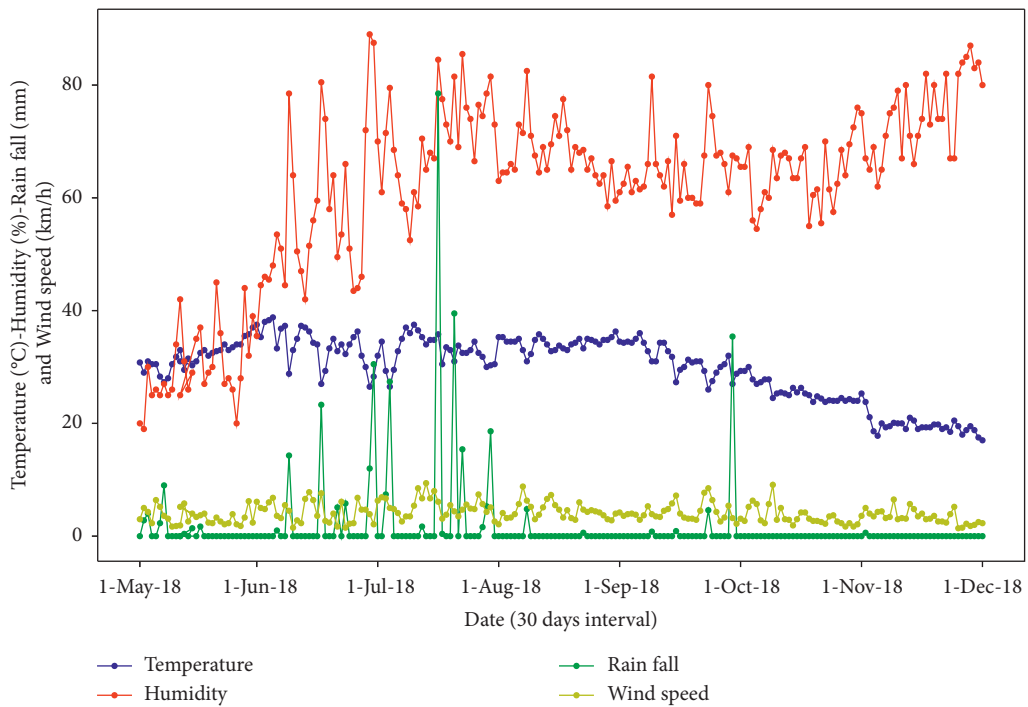


FIGURE 16: Daily average temperature, humidity, rainfall, and wind speed from May to Nov. 2018.

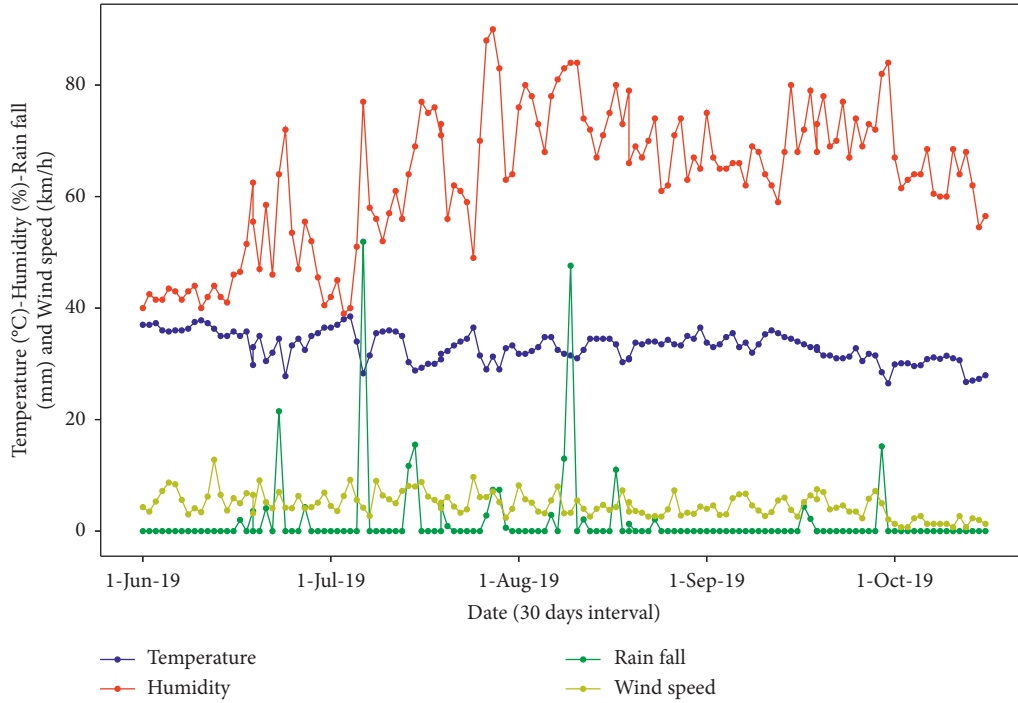


FIGURE 17: Daily average temperature, humidity, rainfall and wind speed from May to Nov. 2019.

TABLE 7: The evaluation metrics of results is precision, recall, F_1 , and support.

| Class | F_1 | Recall | Precision | Support |
|---------------|-------|--------|-----------|---------|
| 0.0 | 0.51 | 0.37 | 0.83 | 27 |
| 1.0 | 0.90 | 0.98 | 0.83 | 84 |
| Macro avg. | 0.70 | 0.67 | 0.83 | 111 |
| Weighted avg. | 0.80 | 0.83 | 0.83 | 111 |

TABLE 8: Cohen’s kappa, ROC AUC, log loss, and confusion matrix measure for predictive features.

| Class | Cohen’s kappa | ROC AUC | Log loss | Confusion matrix |
|--------------|---------------|----------|----------|-----------------------|
| Binary class | 0.427058 | 0.862434 | 0.38 | [[10 17] [2 82]] |

The average 73.5% hotspots of whitefly above ETL have been observed during field evaluation in the experimental area in 2018 and 2019, while on the other, the deep neural network

prediction of whitefly accuracy of 82.88% has been observed. Efficient monitoring of environmental parameters is important for effective prediction processes to achieve the desired results.

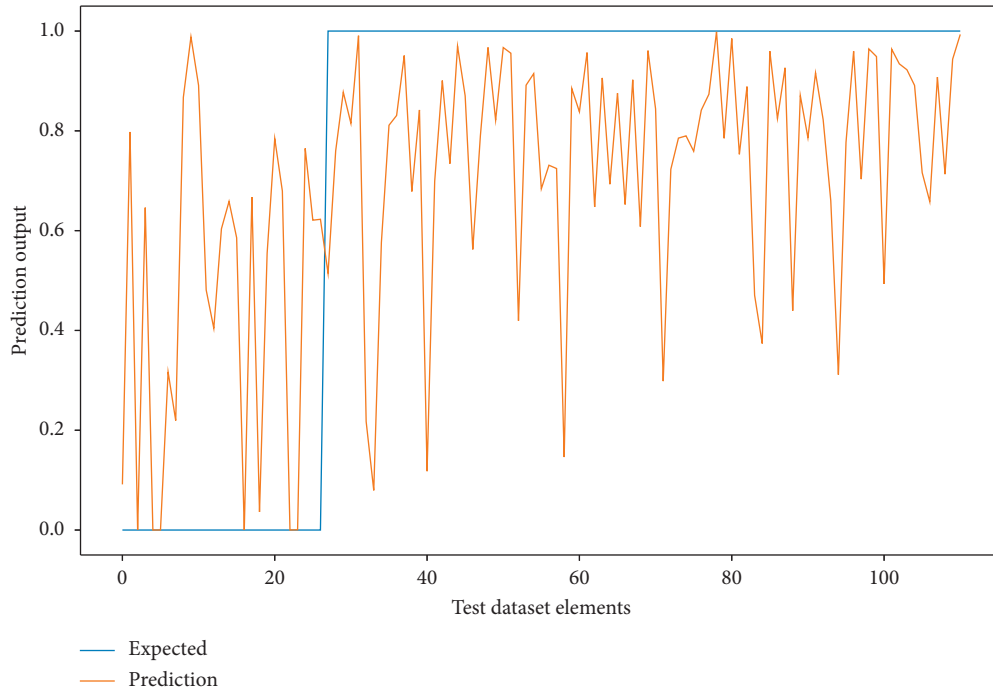


FIGURE 18: Actual and predicted values of test dataset 2018-2019.

TABLE 9: Population of whitefly in percentage.

| Insect pest (white fly) intensity | | Population (%) |
|-----------------------------------|--|----------------|
| Pest population | | |
| 26–40 per leaf | | 76–100% |
| 16–25 per leaf | | 51–75% |
| 11–15 per leaf | | 26–50% |
| 5–10 per leaf | | 1–25% |
| 0–4 per leaf | | 0% |

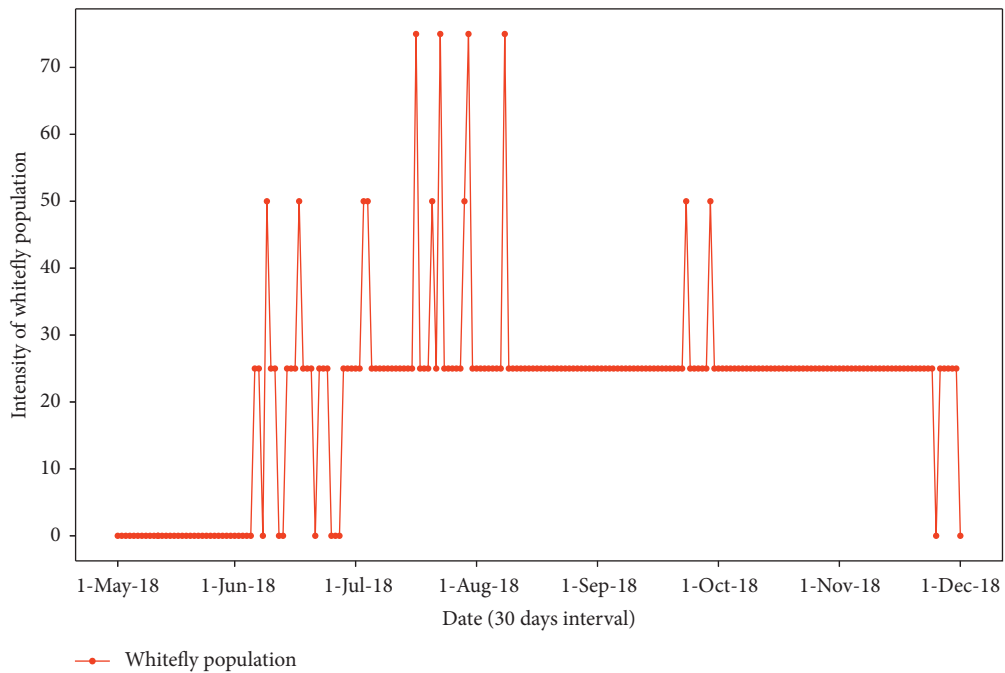


FIGURE 19: Population of whitefly May to Nov. 2018.

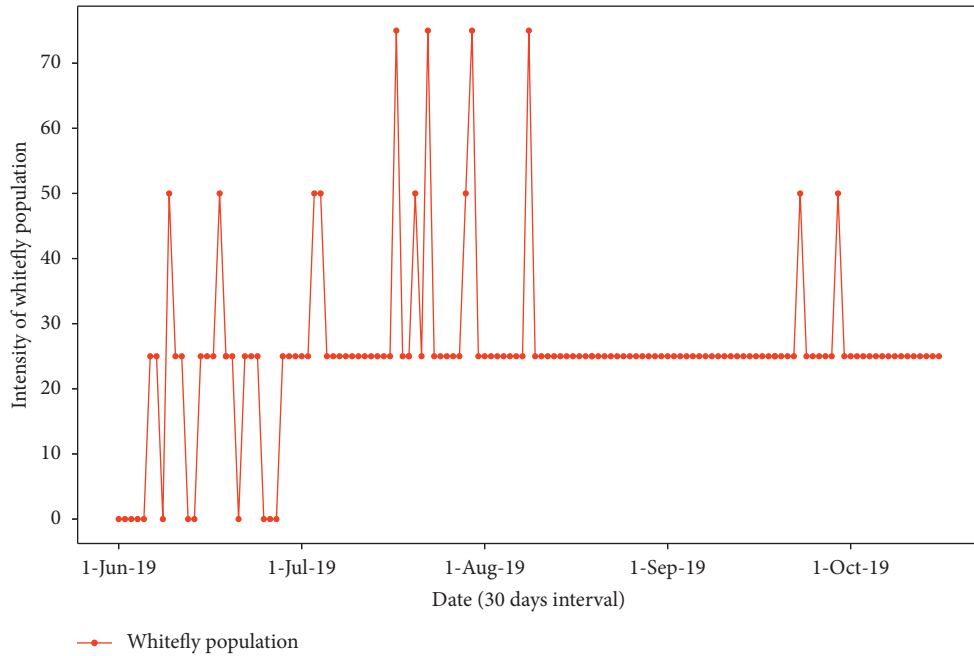


FIGURE 20: Population of whitefly from May to Nov. 2019.

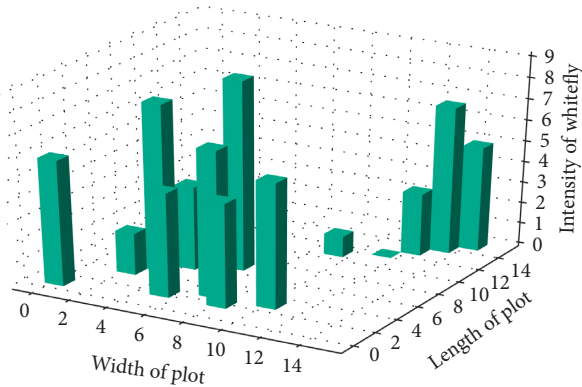


FIGURE 21: Hotspots of whitefly in the experimental area in 2018.

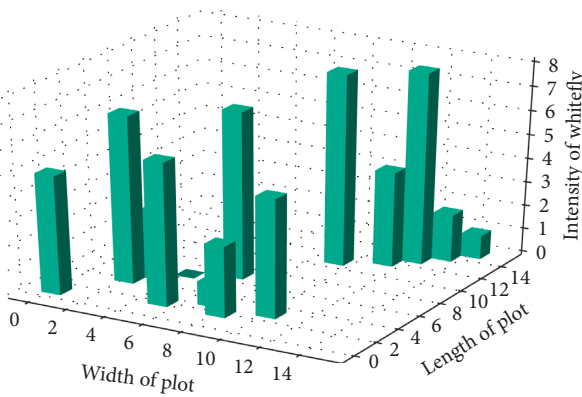


FIGURE 22: Hotspots of whitefly in the experimental area in 2019.

5.3. *Limitation of the Study.* The prediction is based on the abiotic factor only. Many other factors such as the number of host plants, area of host plant cultivation, presence of

predators, and pesticide usage affect the pest population. The factors are important but out of the scope of the study.

6. Conclusion and Future Work

The IoT-assisted crop field context in terms of temperature, humidity, rainfall, and wind speed is used to monitor the environment daily to predict the whitefly attack to take necessary action for the adoption of control measures. The necessities are upgraded by the predominant conditions in the field to effectively serve the target of prediction. These predictions are used to train and test the RBFN algorithm for the deep learning model to optimize these predictions to prevent the attack of whitefly in the cotton crop field. The prediction proved to be very effective regarding the recommendation of pesticides. The prediction of real-time sensor-based environmental data i.e., temperature, humidity, rainfall, and wind speed helps to increase the yield of the crop with high accuracy. The implementation of the suggested model shows major developments in controlling the whitefly attack on the cotton crop area.

The pest prediction needs to be evaluated for other types of pests on different other crops. The inclusion of other factors of pest predictions such as the area of host crops available for the pest can significantly improve the accuracy of the pest prediction mode. Different other biotic and abiotic factors can improve the accuracy of the pest prediction model.

Data Availability

The data used to support the findings of the study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

References

- [1] T. Raney, *The State of Food and Agriculture: Livestock in the Balance*, Food and Agriculture Organization of the United Nations, Rome, Italy, 2009.
- [2] O. Elijah, R. Abdul, I. Orikumhi, C. Y. Leow, and M. H. D. Nour Hindia, "An overview of Internet of Things (IoT) and data analytics in agriculture: benefits and challenges," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3758–3773, 2018.
- [3] X. Shi, X. S. An, Q. Zhao et al., "State-of-the-art internet of things in protected agriculture," *Sensors*, vol. 19, no. 8, p. 1833, 2019.
- [4] A. Tzounis, N. Katsoulas, T. Bartzanas, and C. Kittas, "Internet of things in agriculture, recent advances and future challenges," *Biosystems Engineering*, vol. 164, pp. 31–48, 2017.
- [5] index Mundi, 2019, <https://www.indexmundi.com/agriculture/?commodity=cotton&graph=production>.
- [6] M. A. Ali, *Cotton Production in Pakistan*, John Wiley & Sons, Hoboken, NJ, USA, 2019.
- [7] X.-M. Zhang, G. L. Lovel, M. Ferrante, N.-W. Yang, and F. H. Wan, "The potential of trap and barrier cropping to decrease densities of the whitefly Bemisia tabaci MED on cotton in China," *Pest Management Science*, vol. 76, no. 1, 2019.
- [8] O. Z. Aregbesola, J. P. Legg, L. Sigsgaard, O. S. Lund, and R. Carmelo, "Potential impact of climate change on whiteflies and implications for the spread of vectored viruses," *Journal of Pest Science*, vol. 92, no. 2, pp. 381–392, 2019.
- [9] N. Ahmed, D. De, and I. Hussain, "Internet of things (IoT) for smart precision agriculture and farming in rural areas," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4890–4899, 2018.
- [10] N. Materne and M. Inoue, "IoT monitoring system for early detection of agricultural pests and diseases," in *Proceedings of the 12th South East Asian Technical University Consortium Symposium, SEATUC 2018*, Institute of Electrical and Electronics Engineers Inc, Yogyakarta, Indonesia, March 2018.
- [11] H. Saeed, M. Ehetisham UI Haq, M. Atiq et al., "Prediction of cotton leaf curl virus disease and its management through resistant germplasm and bio-products," *Archives of Phytopathology and Plant Protection*, vol. 51, no. 3-4, pp. 170–186, 2018.
- [12] K. Raghavendra, "Weather based prediction of pests in cotton," in *Proceeding of the International Conference on Computational Intelligence and Communication Networks*, Bhopal, India, November 2014.
- [13] Y. Shang and Y. Zhu, "Research on intelligent pest prediction of based on improved artificial neural network," in *Proceedings of the Chinese Automation Congress (CAC)*, Xi'an, China, December 2018.
- [14] S. Kim, M. Lee, and C. Shin, "IoT-based strawberry disease prediction system for smart farming," *Sensors*, vol. 18, no. 11, 2018.
- [15] S. Ali, Y. Liu, M. Ishaq et al., "Climate change and its impact on the yield of major food crops: evidence from Pakistan," *Foods*, vol. 6, no. 6, 2017.
- [16] A. Tripathy, A. Jagarlapudi, S. Dhanachandran et al., "Data mining and wireless sensor network for agriculture pest/disease predictions," in *Proceedings of the World Congress on Information and Communication Technologies*, Mumbai, India, December 2011.
- [17] D. P. Rubanga, K. Hatanaka, and S. Shimada, "Development of a simplified smart agriculture system for small-scale greenhouse farming," *Sensors and Materials*, vol. 31, no. 3, pp. 831–843, 2019.
- [18] Q. Wang, Y. Zhang, F. Xie, and X. Wu, "Prediction of rice brown planthoppers based on system dynamics," in *Proceedings of the 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, Zhangjiajie, China, August 2015.
- [19] M. S. Mekala and P. Viswanathan, "CLAY-MIST: IoT-cloud enabled CMM index for smart agriculture monitoring system," *Measurement*, vol. 134, pp. 236–244, 2019.
- [20] R. Trogo, J. B. Ebardaloza, D. J. Sabido, G. Bagtasa, E. Tongson, and O. Balderama, "SMS-based smarter agriculture decision support system for yellow corn farmers in Isabela," in *Proceedings of the 2015 IEEE Canada International Humanitarian Technology Conference (IHTC2015)*, Ottawa, Canada, June 2015.
- [21] S. S. Mathurkar, N. R. Patel, R. B. Laanjewar, and R. S. Somkuwar, "Smart sensors based monitoring system for agriculture using field programmable gate array," in *Proceedings of the International Conference on Circuits, Power and Computing Technologies (ICCPCT-2014)*, Nagercoil, India, March 2014.
- [22] T. Li, J. Yang, X. Peng, Z. Chen, and C. Luo, "Prediction and early warning method for flea beetle based on semi-supervised learning algorithm," in *Proceedings of the 2008 4th International Conference on Natural Computation*, pp. 217–221, Jinan, China, November 2008.
- [23] Y.-I. Wei and F.-y. Lin, "The research of prediction of pests based on fuzzy RBF neural network," in *Proceedings of the 2009 International Conference on Computational Intelligence and Software Engineering*, Wuhan, China, December 2009.
- [24] T. Li, J. Yang, and Z. Chen, "The early warning and prediction method of flea beetle based on maximum likelihood algorithm ensembles," in *Proceedings of the International Conference on Natural Computation*, Yantai, China, August 2010.
- [25] H. Lee, A. Moon, K. Moon, and Y. Lee, "Disease and pest prediction IoT system in orchard: a preliminary study," in *Proceedings of the 9th International Conference on Ubiquitous and Future Networks (ICUFN)*, Milan, Italy, July 2017.
- [26] Y. Li, H. Wang, L. M. Dang, A. Sadeghi-Niaraki, and H. Moon, "Crop pest recognition in natural scenes using convolutional neural networks," *Computers and Electronics in Agriculture*, vol. 169, Article ID 105174, 2020.
- [27] J. Liu and X. Wang, "Tomato diseases and pests detection based on improved Yolo V3 convolutional neural network," *Frontiers of Plant Science*, vol. 11, p. 898, 2020.
- [28] Q. Xiao, W. Li, Y. Kai, P. Chen, J. Zheng, and B. Wang, "Occurrence prediction of pests and diseases in cotton on the basis of weather factors by long short term memory network," *BMC Bioinformatics*, vol. 20, no. 25, pp. 1–15, 2019.

- [29] M. Türkoğlu and D. Hanbay, "Plant disease and pest detection using deep learning-based features," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 27, no. 3, pp. 1636–1651, 2019.
- [30] Y. He, H. Zeng, Y. Fan, S. Ji, and J. Wu, "Application of deep learning in integrated pest management: a real-time system for detection and diagnosis of oilseed rape pests," *Mobile Information Systems*, vol. 2019, Article ID 4570808, 13 pages, 2019.
- [31] L.-L. Pan, X.-Y. Cui, Q.-F. Chen, X.-W. Wang, and S.-S. Liu, "Cotton leaf curl disease: which whitefly is the vector?" *Phytopathology*, vol. 108, no. 10, pp. 1172–1183, 2018.
- [32] Pakistan, *Economic Threshold Levels of Insect Pests*, Pest Warning & Quality Control of Pesticides, Government of the Punjab, Lahore, Pakistan, 2018.

Research Article

Clone Chaotic Parallel Evolutionary Algorithm for Low-Energy Clustering in High-Density Wireless Sensor Networks

Rui Yang , Mengying Xu , and Jie Zhou 

College of Information Science and Technology, Shihezi University, Shihezi 8320003, Xinjiang, China

Correspondence should be addressed to Jie Zhou; jiezhou@shzu.edu.cn

Received 14 October 2020; Revised 15 March 2021; Accepted 23 April 2021; Published 29 April 2021

Academic Editor: Wei Li

Copyright © 2021 Rui Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Because the sensors are constrained in energy capabilities, low-energy clustering has become a challenging problem in high-density wireless sensor networks (HDWSNs). Usually, sensor nodes tend to be tiny devices along with constrained clustering abilities. To have a low communication energy consumption, a low-energy clustering scheme should be designed properly. In this work, a new cloned chaotic parallel evolution algorithm (CCPEA) is proposed, and a low-energy clustering model is established to lower the communication energy consumption of HDWSNs. By introducing CCPEA into the low-energy clustering, an objective function is designed for evaluating the communication energy consumption. For this problem, we define a clone operator to minimize the communication energy consumption of HDWSNs, use the chaotic operator to randomly generate the initial population to expand the search range to avoid local optimization, and find the parallel operator to speed up the convergence speed. In the experiment, the effect of CCPEA is compared to heuristic approaches of particle swarm optimization (PSO) and simulated annealing (SA) for the HDWSNs with different numbers of sensors. Simulation experiments demonstrate that the presented CCPEA method achieves a lower communication energy consumption and faster convergence speed than PSO and SA.

1. Introduction

In recent years, because of technological innovation and progress, the volume of microsensor devices has reached the size of a grain of sand. The reduction in volume has made the functions of large-scale wireless sensor networks more perfect, and the cost has also been greatly reduced [1, 2]. Besides, with the development of wireless communication technology and distributed wireless sensor network technology, the high-density wireless sensor networks (HDWSNs) with a large number of nodes, which are densely distributed, gradually become a hot research topic. The HDWSNs are usually made up of a great number of microsensor nodes distributed in the surveillance region at random without any infrastructure support and are self-organized into clusters [3, 4]. The high-density wireless sensor network generally refers to a network in which a large number of wireless sensor nodes are arranged in a small geographic area to achieve a dense perception of targets. At present, this kind of network mostly adopts tree-

like and star-like structures and seldom arranges the aggregation node; the sensor node can reach the base station directly or through a few hops. Due to the high deployment efficiency of high-density sensors and strong environmental adaptability, they profoundly impact many fields, such as national defense and military, smart home, agricultural engineering, environmental monitoring, and many other fields [5, 6].

The HDWSNs are widely used in remote atmospheric monitoring, seismic, radiation, and medical data collection due to their outstanding advantages in information quality, network robustness, network cost, and network adaptability [7, 8]. The HDWSNs are generally self-organizing networks but have different design goals from traditional mobile ad hoc networks. The latter maximizes bandwidth utilization by optimizing routing and resource management strategies in a highly mobile environment while providing users with certain service quality assurance [9]. Most nodes in HDWSNs are static, and there are only a few special nodes that may move. The change of network topology generally

originates from the demise of node energy exhaustion or the demise of node caused by other external reasons [10, 11].

In HDWSNs, sensor nodes are usually randomly dropped by drones to the target area [12]. Although the deployment is simple, the randomness of the deployment cannot guarantee the rationality of the distribution distance of the sensor nodes. Besides, the communication distance between the sensor nodes is also limited, which leads to wasted power consumption for sensor node communication [13, 14]. Therefore, how to select cluster head nodes for clustering in large-scale and high-density sensor networks, which can reduce sensor energy consumption while ensuring the completion of detection tasks and improve the life cycle of sensor networks, is a crucial problem within the research of high-density sensor networks [15, 16]. Due to their size limitation, small wireless sensors have constrained clustering ability [17]. Considering that the power supply capability of sensors is limited, low-energy clusters play a vital role in minimizing communication energy consumption, while most research for low-power clustering is corresponding to a low-power clustering algorithm [18, 19].

In [20], a shuffled frog leaping algorithm (SFLA) technique is presented to search the communication energy consumption in wireless sensor networks (WSNs). SFLA can get lower communication energy consumption than the genetic algorithm (GA). The SFLA technique is simple and fast but suffers from premature convergence. A simple low-energy clustering technique based on quantum genetic algorithm (QGA) while simultaneously evaluating the communication energy consumption to obtain a lower communication energy consumption has also been attempted in [21]. The represented method has presented good results in terms of low energy and communication energy consumption. The QGA is flexible but suffers from the problem of high computational complexity. A low-energy clustering model is suggested in [22] and particle swarm optimization (PSO) is used to resolve the problem. This algorithm has been shown to perform well with a small number of sensors. However, the PSO approach cannot quickly solve the problem while incurring high computational costs.

Based on the concepts and principles of parallel and chaos theory, the new CCPEA is presented in this article. Compared to the traditional evolutionary algorithms (EAs), this technology achieves a better balance and better results. Generally speaking, the hallmark of the CCPEA is a simple heuristic with a good equilibrium mechanism that can flexibly expand and adapt to global and local intelligence capabilities, which has attracted widespread research attention.

In this study, the minimal power clustering issue for low-power WSNs is formulated as a combinatorial optimization problem, taking into account the constraints of energy and monitored area, which is an NP-hard problem. However, it is impossible to perform detailed searches in real-time in HDWSNs [23, 24]. Therefore, numerous heuristic algorithms have been created to reduce WSNs communication power consumption and improve WSNs performance [25–31].

In this article, the clustering problem is transformed into an evolution problem and then solved by the CCPEA. First, we design a new formula for the goal function to match low-power consumption. Furthermore, two new operators, the clone and chaotic operators, are constructed to lower the communication energy consumption in HDWSNs. CCPEA uses powerful parallel operators to mix the advantages of clone selection and chaos generation to solve low-energy clustering problems. We also construct a clone selection to avoid local optima.

Simulations are carried out to denote a comparison of CCPEA through the other two algorithms. From the simulation results, we can get the following conclusions:

- (1) Firstly, the CCPEA can resolve the low-energy clustering problem with lower communication energy consumption than PSO and simulated annealing (SA) techniques. For example, when the number of sensor nodes is 100 and the cluster head ratio is 10%, the energy consumption reduction of CCPEA is 8.46% and 18.55% lower than PSO and SA, respectively.
- (2) Secondly, the CCPEA combines the advantages of the clone operator and the chaotic operator, avoiding the premature convergence problem of PSO and SA. The simulation results show that when the cluster head ratio is 10% and the number of sensor nodes is 300 and 400, the convergence speed of CCPEA is significantly higher than that of PSO and SA.
- (3) Finally, the overall energy loss of the sensor network depends on the sum of the energy loss of the sensor nodes transmitting data and receiving data. As the number of nodes increases, CCPEA can still achieve lower communication energy consumption than PSO and SA technologies while taking the same computational complexity.

2. Related Work

HDWSNs are widely used due to their easy deployment and strong environmental adaptability. Literature [32] installed a large number of wireless sensor nodes on the car and constructed a unique and novel vehicle self-organizing network. The in-vehicle network can analyze the data perceived by the nodes to obtain the driving behavior of the driver and finally give the corresponding insurance level. In [33], the author presents a large-scale high-density wireless sensor network for monitoring the temperature in central Tokyo. The system has a total of 200 sensor nodes arranged in eight monitoring areas, with a node density of approximately 1,800 per square kilometer.

In many industrial applications, it is an important problem to optimize energy by using intelligent algorithm [34–37]. In HDWSNs, an effective low-energy clustering scheme can achieve lower energy consumption, reduce energy costs, and extend network life. For heterogeneous WSNs, literature [38] suggested a brand new distributed low-energy node protection time-driven clustering algorithm (LEPTC) to ensure more uniform energy

consumption of nodes, thereby reducing energy consumption and extending network life. In this algorithm, initialization is performed according to the energy level.

In [39], the author proposed an energy-efficient distributed clustering algorithm in the coverage area. This algorithm considers the redundancy of coverage and the remaining energy of nodes, making the distribution of cluster heads more reasonable. Facts have proved that this algorithm can achieve lower network energy consumption and higher coverage quality.

In [40], the authors proposed a cluster-based routing algorithm in wireless sensor networks based on the genetic algorithm. This algorithm quickly reorganizes clusters in a network with uneven distribution of nodes and selects new cluster heads to achieve a balance of energy consumption, thereby achieving a longer network life.

In [41], a method based on the energy-efficient genetic algorithm is proposed, which improves the overall performance of WSNs based on the Virtual Grid-Based Dynamic Routes Adjustment (VGDR). Compared with other methods, this dynamic method better balances the load and optimizes it, thereby creating more opportunities and achieving better results with fewer loops.

In literature [42], the author proposed a multiobjective Bat algorithm to find the best cluster formation in WSNs and proposed a routing model. The optimal node is used as the cluster head and the communication distance is modeled by Bat loudness parameters to optimize the energy consumption in WSNs.

According to the characteristics of WSNs, the study in [43] suggested a routing algorithm for WSNs based on ant colony optimization. The outcomes demonstrate that the enhanced scheme has good performance in terms of power consumption and global optimization capabilities.

Aiming at the defect of premature convergence of the traditional K-means clustering algorithm, the article [44] proposed an improved GA based on the hybrid K-means clustering algorithm, which can prevent the algorithm from falling into the local optimum by introducing an adaptive function.

The study in [45] proposed a hybrid method called KGA, which aims to combine GA and K-means algorithm to search for the optimal number of clusters, thereby optimizing the communication energy in WSNs.

In [46], low-energy clustering problem approach for low-energy clustering problems in WSNs to minimize the communication energy consumption is researched based on PSO. In their article, they minimize the communication energy consumption without considering energy restriction. However, it also suffers from an excessive computational time requirement.

In [47], clustering design techniques based on SA have been represented in order to maximize the network lifespan in a long network lifespan. Their design is a similar concept to the GA. The SA approach is simple and fast but suffers from premature convergence.

In [48], the clustering design strategy for clustering design in WSNs to maximize the network lifespan is explored based on the quantum evolutionary algorithm

(QGA). The QGA method employs an individual to suggest the solution and obtains the longer network lifespan iteratively. QGA performs well in the beginning, but it suffers from premature convergence and a low convergence rate only after a few iterations.

3. System Model

In this section, a low-energy clustering model is proposed for the constraints of sensor node power and communication energy consumption in HDWSNs. The typical network structure diagram of HDWSNs studied in this article is shown in Figure 1:

As shown in Figure 1, in the HDWSNs, the clustering structure means that a cluster head node will be chosen from a similar area within the monitoring range, and a node cluster will be formed around the cluster head node. Each sensing node perceives the target and then uploads the sensing result to the primary cluster node in the node cluster after completing the sensing task. The cluster head node collects the sensing results uploaded by the sensing nodes in the cluster and then directly uploads the results to the gateway node in multiple hops. The monitoring task performed by the sensing node is that the gateway node publishes the task to the cluster head node; after that, the cluster head node distributes the task to the sensing nodes in the cluster.

This article mainly studies how to minimize the power consumption of HDWSNs by optimizing the energy consumption of communication between nodes through reasonable clustering. As the communication distance between sensor nodes is limited, the communication power consumption of the sensor network when sending and receiving data cause serious waste of energy [17, 49, 50]. Therefore, it is extremely important to develop a reasonable and efficient low-energy clustering scheme. According to the low-energy clustering model in literature [51], the formula for the energy consumed by the sending node to transmit b bits of data to the receiving node can be obtained by the following formula:

$$\text{cost}_{\text{send}}(b, l) = E_{\text{elec}} \cdot b + \epsilon_{\text{amp}} \cdot b \cdot l^n. \quad (1)$$

In formula (1), $\text{cost}_{\text{send}}$ represents the energy consumed when the node sends b bits data to the receiving node and the distance between the two nodes is l . Among them, E_{elec} represents the electronic energy parameter, ϵ_{amp} represents the power amplification parameter, and the value of n , usually between 2 and 4, is generally determined according to the quality of the communication environment. The better the communication environment, the smaller the value of n . At the same time, the communication energy required by the receiving node to receive b bits data is shown as follows:

$$\text{cost}_{\text{received}}(b) = E_{\text{elec}} \cdot b. \quad (2)$$

In formula (2), $\text{cost}_{\text{received}}$ indicates the communication energy required by the receiving node to receive b bits energy. In the model of this article, suppose b is 1 M bits, $\epsilon_{\text{amp}} = 100 \text{ pJ/bit/m}^2$, $E_{\text{elec}} = 50 \text{ nJ/bit}$, $n = 3$.

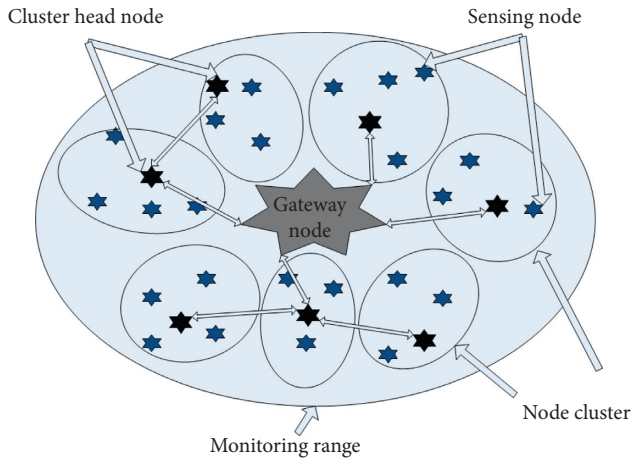


FIGURE 1: Cluster structure of density sensor network.

4. CCPEA-Based Low-Energy Clustering Problem in HDWSNs

The EA is one of the most popular metaheuristic algorithms, which attempts to mimic the procedure of natural selection [19]. It is also an exploit method that mimics the procedure of natural selection in nature, which is an optimization algorithm to search an input region while minimizing a result function under given restraints [23, 52, 53]. The primary thought is to get inspired analogy from the natural mechanisms of gene recombination and mutation. There exist multiple alternatives to implementation for heuristic operators. EA is researched as a suitable metaheuristic, usually used to settle complicated optimization issues [54].

Inspired by traditional EAs, this section describes the design of the CCPEA for the low-power clustering problem. In this article, a novel clone method has been studied to minimize the communication energy consumption in HDWSNs. Furthermore, two novel procedures, the chaotic and parallel procedures, are formed. On the one hand, we reveal that a chaotic procedure depending on the binary region can be naturally integrated with EA so that feasible solutions are completely searched. On the other hand, with the parallel operator, it is more effective for the operator of diverse lengths of chromosomes compared to the traditional EA. In CCPEA, clone, chaotic and parallel procedures are helpful to enhance the population diversity of CCPEA and avoid premature convergence.

Therefore, the depicted CCPEA procedures are repeated at a time granularity stationary by HDWSNs requirements. The CCPEA employs various simple procedures in order to simulate evolution. So, the suggested CCPEA-based clustering problem can be summarized as follows:

- (i) Initialize the chromosomes of CCPEA
- (ii) Select superior chromosomes as parents to feed into the genetic procedure
- (iii) Generate a novel population by crossover and mutation

- (iv) Then, their result function value with the result function is evaluated
- (v) Update the population by switching inferior chromosomes

The loop is repeated until the stopping condition is satisfied. The comprehensive description of the CCPEA utilized to explore the almost best clustering problem is defined in what follows.

4.1. Representation of Chromosomes. The first step to design CCPEA is to find a suitable chromosome representation scheme. The efficiency of a CCPEA depends on the encoding technique employed. In CCPEA, a solution is presented by a chromosome. For a low-energy clustering problem, each chromosome in the group might imply a collection of randomly selected clustering problems. For the scope of this article, the variable should be suggested by a binary code representing the clustering problem for the low-energy clustering problem. The Boolean coding representation is correct and powerful because it is closest to the clustering problem, and the string length is the number of sensors. A chromosome is composed of a string of binary symbols. By doing this, each chromosome is converted from the Boolean string into real numbers to gain the communication energy consumption associated with each member of the group. Each chromosome is made up of bits. CCPEA is easy to use since there are only two options to utilize to a bit: 0 or 1.

If the code of a chromosome is "0100110101," the number of genes in the chromosome is 10, and each gene represents a sensor node; that is, the quality of sensor nodes in the sensor network is 10. In other words, the chromosome symbol length is the number of sensors in the HDWSNs. When the gene at a location is 1, it means that the sensor at that location is a cluster head node, and 0 is a sensory node. For example, if the second digit of the chromosome is 1, the second sensor node is the cluster head node.

4.2. Initial Population. The CCPEA requires a group of potential solutions to be initialized at the beginning of the CCPEA procedure. The CCPEA solves the optimization problem by manipulating a group of chromosomes. CCPEA solves optimizing issues according to a group of a stationary number, referred to as the group size, of solutions. Generally, in the case of a very small population, only a small part of the exploited area is explored, thereby increasing the risk of premature convergence to local extremes. It keeps a group of chromosomes that evolves over successive generations. In CCPEA, a group is randomly created. In finding bits that satisfy constraints, CCPEA applies a random number generator. A random initial group is generated as a group of solutions of clustering problems. In this article, the size of the initial population is set to S , and there are N genes in the initial population; that is, there are N sensor nodes in the HDWSNs, and the amount of cluster heads is fixed to M . The population coding can be described as

$$F = \begin{bmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,N-1} & q_{1,N} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,N-1} & q_{2,N} \\ \vdots & \vdots & & \vdots & \vdots \\ q_{S-1,1} & q_{S-1,2} & \cdots & q_{S-1,N-1} & q_{S-1,N} \\ q_{S,1} & q_{S,2} & \cdots & q_{S,N-1} & q_{S,N} \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_{S-1} \\ Q_S \end{bmatrix} \quad (q_{s,n} \in \{0,1\}), \quad (3)$$

$$\sum_{n=1}^N q_{s,n} = M \quad (s \in \{1, 2, \dots, S\}). \quad (4)$$

In formula (3), Q_s is expressed as the s_{th} individual, and whether the n_{th} sensor in the s_{th} individual is a cluster head node is expressed by the value of $q_{s,n}$. If $q_{s,n} = 1$, it means that the sensor node is a cluster head node; otherwise, it is a sensing node. Formula (4) is the constraint on the number of cluster head nodes in each individual, and the fixed value is M . For example, when the number of sensors in a certain individual is 100, then $N = 100$. If the cluster head ratio is selected as 10%, then M is 10.

4.3. Fitness Evaluation. In the given context, we calculate the result function value of each chromosome according to the communication energy consumption. The purpose of this article is to minimize communication energy consumption. In this way, the value of fitness is lower, the quality of the chromosomes is better. CCPEA solves stochastic optimization problems, and each chromosome is evaluated by a fitness function. The fitness function blends the satisfaction of restrictions by handling the clustering design problem. For an individual in the CCPEA, its fitness value is calculated based on the fitness function. In this article, the fitness value of the chromosome is calculated by (5). In CCPEA, we calculate the communication energy consumption of each chromosome's fitness value. Therefore, in terms of fitness value, our aim is to find a solution using

$$\text{Fit}(Q) = \sum_{n=1}^N (\text{cost}_{\text{send}} + \text{cost}_{\text{received}}). \quad (5)$$

4.4. Selection. The CCPEA adopts a roulette wheel selection strategy to select the parents. The selection procedure adopts the best preservation approach and roulette technique. In the roulette wheel, the individual with better result function values has more possibility of being selected. Thus, it is possible to select the same chromosome at various times. According to this selection probability, a pair of parent solutions is chosen from the current population. The algorithm then replaces low-quality solutions through recently developed high-quality solutions to obtain a much better current group. The survived chromosomes are then utilized to produce the next iteration. In this way, chromosomes that are more common in the survived population are more likely to be inherited. The replaced individual is chosen randomly, but whose result function value must be above the average level in the community.

In order to ensure that individuals with lower communication energy consumption get a greater probability of being chosen, the probability of individual Q_s , that is, the possibility of an individual being chosen, is inversely proportional to the degree of fitness, as shown in

$$F_{\text{SELECT}}(Q_s) = \frac{(1/\text{Fit}(Q_s))}{\sum_{i=1}^S 1/\text{Fit}(Q_s)}. \quad (6)$$

4.5. Crossover. By the chance of selection, the chosen chromosomes are directly transferred to the crossover. Crossover is to find a better solution to deal with the current solution. The crossover is an operation carried out to produce offspring by taking characteristics from the parents. Crossover is a heuristic procedure for recombining two parent solutions into two new solutions.

According to the literature [55], the concept of GA crossover, assuming that the two individuals Q_1 and Q_2 are cross-operated, Q' is first obtained through the logical AND operation. Q' is to compare the Boolean algebras of the corresponding positions in the two individuals. If the Boolean algebra of the corresponding position is the same, it remains unchanged, and if it is different, it becomes 0, as shown in Figure 2.

Secondly, perform the logical AND operation on the two individuals to get Q'' . Q'' changes the positions of the Boolean algebra of the corresponding positions in Q_1 and Q_2 to 0, and the difference to 1 as shown in Figure 3.

Finally, evenly distribute the '1' in the position of Q'' to the corresponding position in Q' to obtain two new individuals generated by crossover. As shown in Figure 4, the number of '1' in Q'' is evenly allocated to Q' , and the position is random. Therefore, the two newly obtained individuals can be $Q_{\text{new1}} = [1000101001]$ and $Q_{\text{new2}} = [1000100110]$.

4.6. Mutation. Each chromosome had a given possibility of being mutated; for the CEAEA, this probability is defined to 0.05. The main goal of the mutation program is to maintain diversity within the group. Considering that the amount of cluster heads in an individual is constant, the mutation operation randomly changes a position of "1" in the individual to "0" with a mutation probability and randomly selects one of the positions where the value is "0" and changes it to "1," as shown in Figure 5.

4.7. Clone. The cloning algorithm is an optimized algorithm inspired by the cloning principle of the biological immune system. The cloning algorithm combines the adaptive ability of the biological immune system with the prior knowledge of the problem, so the algorithm has good robustness in the information search process and guides the search process to converge in the direction of the global optimal solution. The CCPEA algorithm increases the population size through the cloning operator, effectively increases the diversity of the population, and helps to find the global optimal solution.

$$\begin{aligned}
Q_1 &= [1\ 0\ 0\ 0\ 1\ 0\ \boxed{0\ 1\ 0\ 1}] \\
Q_2 &= [1\ 0\ 0\ 0\ 1\ 0\ \boxed{1\ 0\ 1\ 0}] \\
Q' &= [1\ 0\ 0\ 0\ 1\ 0\ \boxed{0\ 0\ 0\ 0}]
\end{aligned}$$

FIGURE 2: Crossover.

$$\begin{aligned}
Q_1 &= [1\ 0\ 0\ 0\ 1\ 0\ \boxed{0\ 1\ 0\ 1}] \\
Q_2 &= [1\ 0\ 0\ 0\ 1\ 0\ \boxed{1\ 0\ 1\ 0}] \\
Q'' &= [0\ 0\ 0\ 0\ 0\ 0\ \boxed{1\ 1\ 1\ 1}]
\end{aligned}$$

FIGURE 3: Crossover.

$$\begin{array}{cc}
Q' = [1\ 0\ 0\ 0\ 1\ 0\ \boxed{0\ 0\ 0\ 0}] & Q' = [1\ 0\ 0\ 0\ 1\ 0\ 0\ \boxed{0\ 0\ 0}] \\
Q'' = [0\ 0\ 0\ 0\ 0\ 0\ \boxed{1\ 1\ 1\ 1}] & Q'' = [0\ 0\ 0\ 0\ 0\ 0\ 1\ \boxed{1\ 1}] \\
Q_{new1} = [1\ 0\ 0\ 0\ 1\ 0\ \boxed{1\ 0\ 0\ 1}] & Q_{new2} = [1\ 0\ 0\ 0\ 1\ 0\ 0\ \boxed{1\ 1\ 0}]
\end{array}$$

(a) (b)

FIGURE 4: Crossover.

$$\begin{aligned}
Q_1 &= [1\ 0\ 0\ 0\ \boxed{1}\ 0\ 0\ 1\ \boxed{0}\ 1] \\
Q_2 &= [1\ 0\ 0\ 0\ \boxed{0}\ 0\ 0\ 1\ \boxed{1}\ 1]
\end{aligned}$$

FIGURE 5: Mutation.

4.8. Chaotic. In CCPEA, when the EA initializes the population, it has a greater impact on the iterative optimization of subsequent generations. Therefore, the use of chaotic sequence logistic mapping to improve the evolutionary population can enrich the diversity of the initial population and accelerate the optimization speed. Because chaotic mapping causes chaos in the feasible region of the independent variable, it is predictable in a short initial time, but it is random in a long time. Therefore, chaotic mapping has a positive effect on the convergence speed of EAs.

4.9. Parallel. In CCPEA, a shared area is opened by the main thread to save the optimal individual of each thread. The child threads run their GAs, synchronize their optimal individuals to the shared area every hundred generations, and introduce optimal individuals from other threads.

4.10. Computational Complexity Analysis. In this part, we analyze the computational complexity of the proposed CCPEA. In the low-energy clustering problem, the distance between sensor nodes directly affects the energy cost between sensors, so it is necessary to calculate the distance and energy consumption between each sensor node. In the system model of this article, there are N sensor nodes, so the computational complexity of energy consumption calculation is $O(N^2)$. For CCPEA, there are S individuals in a population, and each contains N sensor nodes. If the number of iterations is H , the computational complexity is $O(N^2) + O(HSN)$.

5. Simulation and Discussion

We propose the simulation results for low-power clustering in HDWSNs with CCPEA, PSO, and SA in this section. Simulations were performed to verify the low-energy clustering performance of the proposed CCPEA method. We test the performance of the schemes on a PC with Intel Core i7-8550 U, 2.00 GHz, 8 GB RAM, Win10 operating system, and MATLAB software to denote its applicability to the clustering design problem. To evaluate the performance of the CCPEA and other heuristics, a single result function, as described in Section 4, is utilized in the experimental results. Then, we develop sensor nodes, and the coordinate of each sensor node is randomly specified within the square region. Four low-power clustering problem cases with diverse numbers of sensors are tested. The performance of the CCPEA, PSO, and SA is reported.

For CCPEA, the selection of parameters is based on the range of empirical values based on existing research, and the parameters are adjusted according to the range of empirical values. Due to the sensitivity of the parameters, slight changes in parameter data will affect the performance of the algorithm. Therefore, many experiments must be carried out and the parameters must be adjusted several times until the algorithm achieves better performance. At present, a simulation model close to reality is used to verify the rationality of the experimental results, which will be implemented in the actual system in the near future.

In our simulation, all comparisons between CCPEA, PSO, and SA were reported using 100 generations and 40 individuals. In CCPEA, using recommendations, we select

0.05 as mutation probability and 0.8 as crossover possibility. The parameter values in the PSO are based on a parametric study, the learning factor $C1 = C2 = 2$ is selected, and the maximum velocity of the particle is fixed to 6. In SA, the initial temperature and annealing temperature coefficients are 200 and 0.85, respectively. The specific description of the parameters is shown in Table 1.

The basic concept taken in this work is as follows: the nodes are connected by wireless communication, and the energy consumption is composed of the sum of the energy consumed by the receiving node and the sending node when sending and receiving energy. In the experiment, in order to consider the influence of the number of different sensor nodes and different cluster head ratios on the experimental results, a large number of simulation experiments were done for the different numbers of sensor nodes and different cluster head ratios, and the following similar situations were obtained. This article mainly focuses on the comparison of the communication energy consumption of the three algorithm schemes when the ratio of cluster heads is 10% and the number of sensor nodes is 100, 200, 300, and 400, respectively. And when the ratio of cluster heads is 5%, 10%, 15%, and 20%, the energy consumption of the three algorithms is compared when the sensor nodes are 200, 400, 600, 800, 1000, and 1200, respectively, as shown in Table 1.

Figures 6(a)–6(d) show the comparison of communication energy consumption optimized by CCPEA, PSO, and SA when the number of sensor nodes is 100, 200, 300, and 400 when the proportion of cluster heads is 10%. For each technique, we just choose the optimum solution in each iteration from the present population. It must be noted that the experiment of CCPEA is superior to the PSO and SA methods, which can be obtained in Figure 6. In Figure 6(a), compared to other techniques, after 100 generations, the communication energy consumption of CCPEA is reduced to 68.25 J. However, PSO and SA attain suboptimal results, and the communication power consumption acquired by the PSO and SA is 74.56 J and 83.79 J, respectively. CCPEA reduces communication energy consumption by 8.46% and 18.55% than PSO and SA.

In Figures 6(b)–6(d), 200, 300, and 400 sensor nodes are used to obtain similar results. In Figure 6(b), the communication energy consumption of CCPEA, PSO, and SA reached 112.52 J, 125.78 J, and 147.72 J, respectively. And compared with PSO, SA, CCPEA achieved a faster convergence rate in the first 50 generations and achieved lower energy consumption in the later 50 generations. In Figure 6(c), the communication energy consumption using the CCPEA method dropped to 162.27 J, while PSO and SA dropped to 184.68 J and 219.44 J, respectively. In Figure 6(d), the communication energy consumption of CCPEA, PSO, and SA is 183.36 J, 207.82 J, and 257.69 J, respectively. The communication energy consumption of CCPEA is reduced by 11.77% and 28.84% lower than that of PSO and SA. And before the 40th generation, the convergence speed of CCPEA was significantly faster than the other two algorithms.

As shown in Figure 6, the value of communication energy consumption initially decreases with the growth of generations. It can be seen that CCPEA finds high-quality

experiments much faster than PSO and SA. On the other hand, the PSO and SA denotes a quite slower convergence, hence proving the superior reliability of CCPEA. CCPEA combines the advantages of the cloning operator, accelerates the convergence speed, has better reliability, and solves the shortcomings of the slow convergence speed of traditional intelligent algorithms. In CCPEA, the cloning operator is used to replicate the 5 best individuals in the population and inherit them to the population in the next generation to ensure the population in the next generation is better than the previous population. Therefore, achieving better convergence can be achieved. It is evident that CCPEA has converged to better solutions and it is prevented from premature convergence. In all 100 generations, the communication energy consumption of CCPEA is lower than that of PSO and SA, and the chaotic operator is used to generate a random initial population, expand the search range of the population, which helps to find a better solution, achieve lower energy consumption, and effectively avoid the algorithm from stagnating early. It can be seen that the solutions found by CCPEA propose stable performance, which denotes the robustness of the algorithm. The simulation results present that the suggested CCPEA method offers lower communication power consumption over the current PSO and SA methods.

Figures 7(a) and 7(b) show a comparison of the communication energy consumption changes of CCPEA, PSO, and SA with different amount of sensor nodes when the proportion of cluster heads is 5%, 10%, 15%, and 20%, respectively. Figure 7(a) illustrates the communication power consumption corresponding to the number of different sensor nodes while the proportion of cluster head nodes is 5%. The specific values can be seen in Table 2.

As shown in Table 2, when the number of sensors is 1200, the optimal communication energy consumption of CCPEA is 361.54 J, while the communication energy consumption obtained by PSO and SA is 389.77 J and 518.82 J, revealing that the CCPEA is more robust than PSO and SA for minimizing the communication energy consumption. The same result can be obtained in Figures 7(b)–7(d).

Figure 7(b) illustrates the communication power consumption corresponding to the number of different sensor nodes while the proportion of cluster head nodes is 10%. It can be seen from Figure 7(b) that when the number of sensor nodes is 1200, the energy consumption cost of CCPEA is 428.42 J, and the energy consumption costs of PSO and SA are 534.56 J and 872.69 J, respectively. The energy consumption cost of CCPEA is much lower than that of PSO and SA.

Figure 7(c) illustrates the communication power consumption corresponding to the number of different sensor nodes while the proportion of cluster head nodes is 15%. When the number of sensor nodes is 1200, the energy consumption cost of CCPEA is 509.53 J, and the energy consumption costs of PSO and SA are 788.24 J and 1277.36 J, respectively. The energy cost of CCPEA is 35.36% and 60.11% lower than that of PSO and SA, respectively.

TABLE 1: Description of parameters.

| Parameters | Description |
|---------------------------------------|--|
| Number of individuals | An individual represents a solution to a low-energy clustering problem |
| Number of iterations | Algorithm optimization times |
| Mutation rate | Probability of binary code mutation |
| Crossover rate | Probability of binary change exchange between two individuals |
| Learning factors C1 and C2 | Acceleration constant, normally, $C1 = C2 = 2$ |
| Maximum velocity of the particle | Maximum speed of particle movement |
| The initial temperature | A sufficiently large temperature defined before the first iteration |
| The annealing temperature coefficient | Cooling rate coefficient, when the cooling rate coefficient is smaller, the cooling rate is faster |

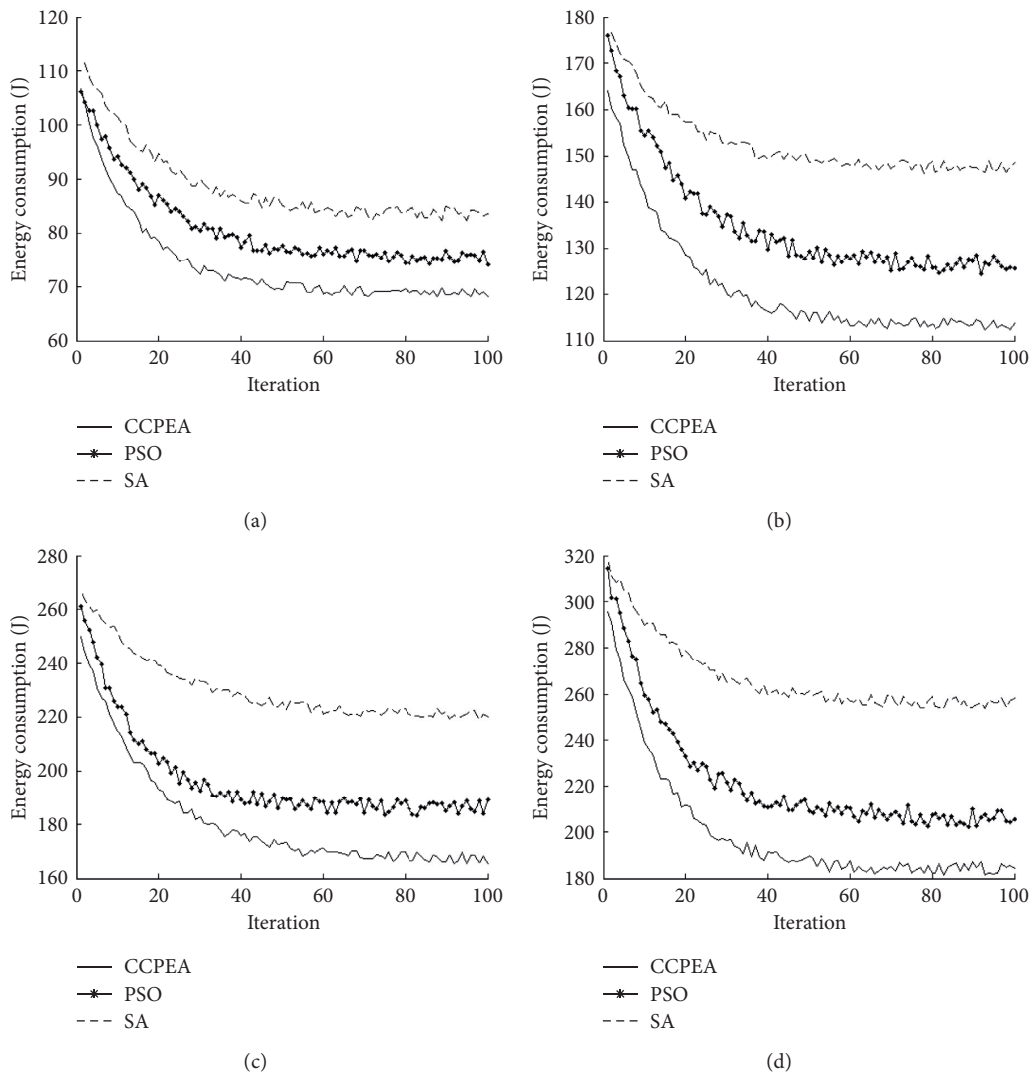


FIGURE 6: Energy consumption of CCPEA, PSO, and SA ((a) 100 sensor nodes; (b) 200 sensor nodes; (c) 300 sensor nodes; (d) 400 sensor nodes).

Figure 7(d) illustrates the communication power consumption corresponding to the number of different sensor nodes while the proportion of cluster head nodes is 20%. When the number of sensor nodes is 1200, the energy consumption cost of CCPEA is 663.72J, and the energy consumption costs of PSO and SA are 1058.47J and

1726.68J, respectively. The energy cost of CCPEA is 37.29% and 61.56% lower than that of PSO and SA, respectively.

When the number of sensor nodes is fixed at 1000, in Figure 8(a)-8(b), the communication energy consumption of the three algorithms at different cluster head ratios is compared. In Figure 8(a), the communication energy of

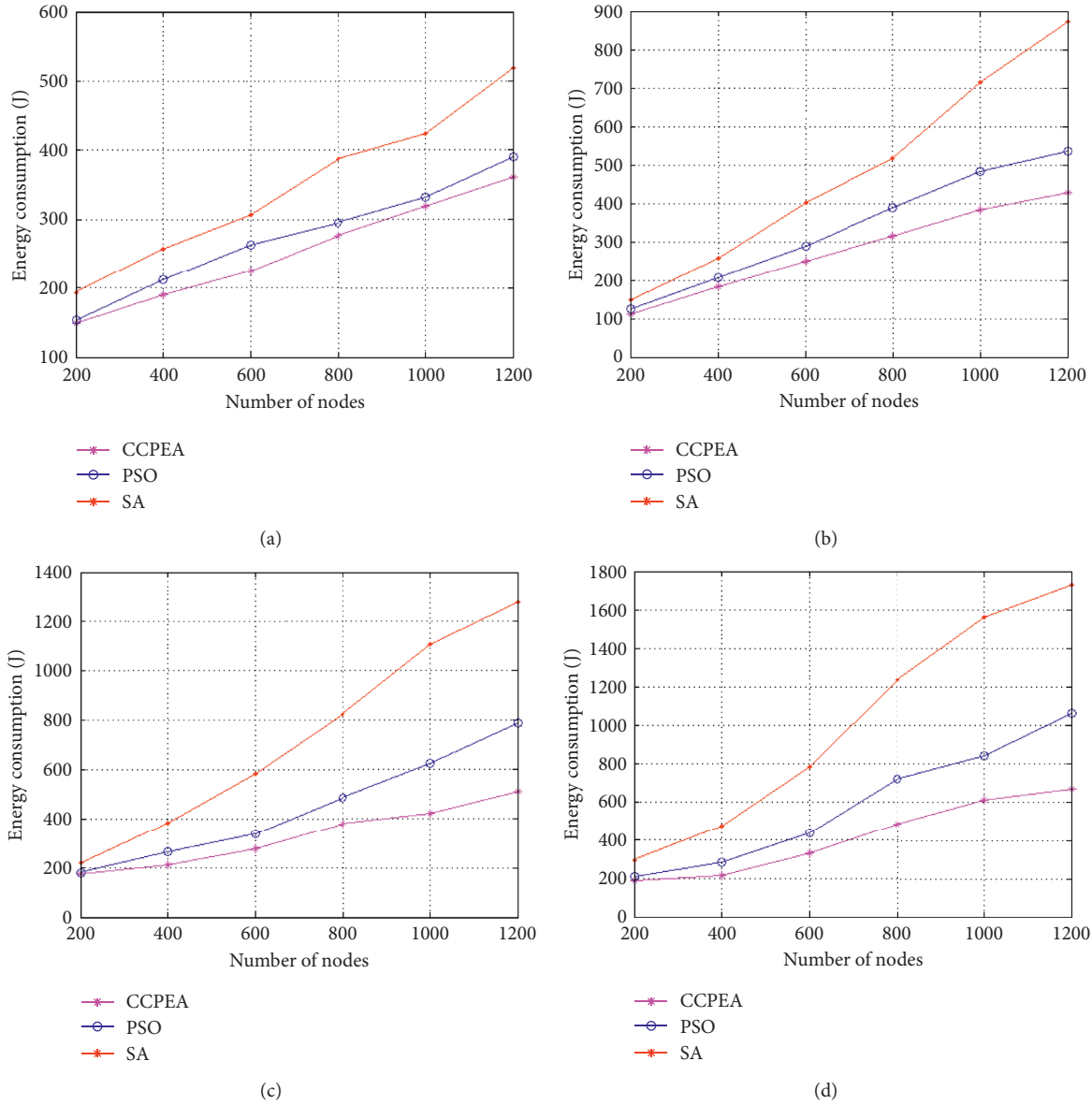


FIGURE 7: Energy consumption of CCPEA, PSO, and SA ((a) 5% cluster head ratio; (b) 10% cluster head ratio; (c) 15% cluster head ratio; (d) 20% cluster head ratio).

TABLE 2: Energy consumption when the cluster head ratio is 5% (J).

| | 200 sensors | 400 sensors | 600 sensors | 800 sensors | 1000 sensors | 1200 sensors |
|-------|-------------|-------------|-------------|-------------|--------------|--------------|
| CCPEA | 148.70 | 189.48 | 224.89 | 276.18 | 318.60 | 361.54 |
| PSO | 152.66 | 211.72 | 263.32 | 295.29 | 331.01 | 389.77 |
| SA | 193.34 | 257.26 | 306.31 | 387.46 | 423.67 | 518.82 |

CCPEA is 318.60 J with a cluster head ratio is 5%, while the low-energy clustering solutions in PSO and SA are 331.01 J and 423.67 J, respectively. In Figure 8(b), when the cluster head ratio is 10%, the communication energy of CCPEA is 381.67 J and PSO and SA are 482.49 J and 715.77 J, respectively. Figure 8(c) represents the communication energy cost of the three algorithms when the cluster head ratio is 15%. The communication energy of CCPEA is 421.46 J, while

that of PSO and SA is 623.13 J and 1108.34 J, respectively. In Figure 8(d), the communication energy of CCPEA is 318.60 J, and those of the PSO and SA are 331.01 J and 423.67 J, respectively. In Figures 8(a)–8(d), we can also clearly conclude that CCPEA is more robust and stable than ACO and SA. Simulation denotes that the suggested CCPEA strategy outperforms the conventional ACO and SA technologies with smaller communication energy consumption.

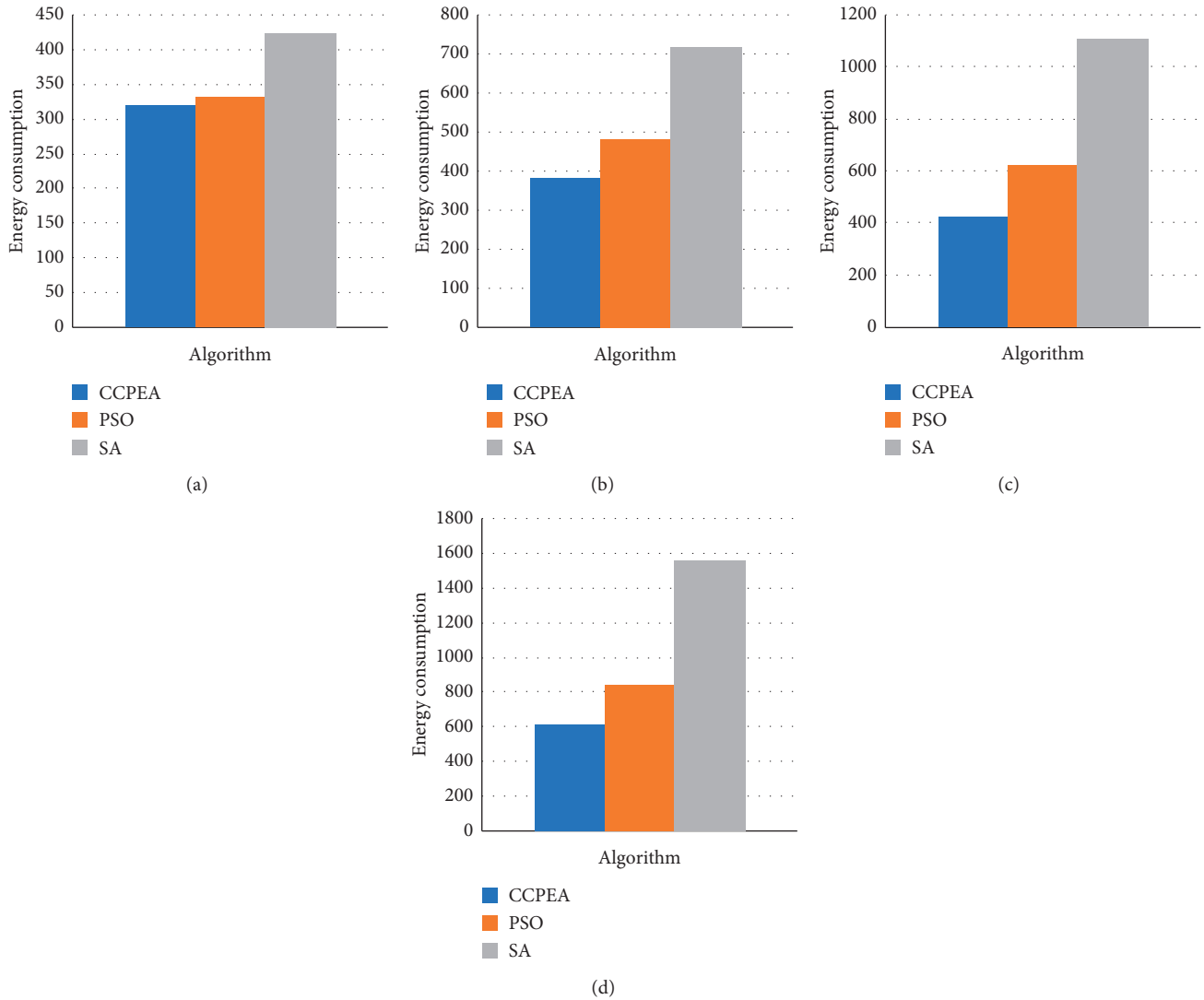


FIGURE 8: Energy consumption when the number of sensor nodes is 1000 ((a) 5% cluster head ratio; (b) 10% cluster head ratio; (c) 15% cluster head ratio; (d) 20% cluster head ratio).

6. Conclusion

This work presents a novel clone chaotic parallel evolutionary algorithm (CCPEA), which uses the merging clone operator and chaotic operator. In this article, we first describe a new formulation of the objective function to minimize the communication energy consumption to suit the low energy. By introducing CCPEA into the low-energy clustering, a result function for evaluating the communication energy consumption is designed to minimize the communication energy consumption for HDWSNs. Comprehensive analysis and experiments are carried out to assess the performance improvement of CCPEA compared to methods according to PSO and SA. The experimental results show that, in the case of different cluster head ratios and different sensor nodes, the communication energy consumption of CCPEA is lower than that of PSO and SA. The cloning operator, chaos operator, and parallel operator in CCPEA expand the scope of optimization and reduce the

energy consumption of communication while avoiding the premature convergence and evolutionary stagnation of the algorithm.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest in this work.

Acknowledgments

This study was supported by the project of Youth and Middle-Aged Scientific and Technological Innovation Leading Talents Program of the Corps (No. 2018CB006),

project of Corps Finance Science and Technology Plan (No. 2020CB001), the second batch of funding project for high-level talents research in Shihezi University in 2018 (No. RCZK2018C38), research on Similarity Recommendation Algorithm of Criminal Cases (No. ZZZC201915B), and postgraduate education innovation program of the Autonomous Region.

References

- [1] W. Jiang, P. Wan, Y. Wang, W. Su, and D. Liang, "A localization algorithm based on the hops for large-scale wireless sensor networks," in *Proceedings of the 2014 International Conference on Wireless Communication and Sensor Network*, pp. 217–221, Wuhan, China, December 2014.
- [2] H. Kim and S.-W. Han, "An efficient sensor deployment scheme for large-scale wireless sensor networks," *IEEE Communications Letters*, vol. 19, no. 1, pp. 98–101, 2015.
- [3] C. Li, J. Li, M. Jafarizadeh, G. Badawy, and R. Zheng, "LEMoNet: low energy wireless sensor network design for data center monitoring," in *Proceedings of the 2019 IFIP Networking Conference (IFIP Networking)*, pp. 1–9, Warsaw, Poland, May 2019.
- [4] O. Banimelhem, M. Naserllah, and A. Abu-Hantash, "An efficient coverage in wireless sensor networks using fuzzy logic-based control for the mobile node movement," in *Proceedings of the 2017 Advances in Wireless and Optical Communications (RTUWO)*, pp. 239–244, Riga, Latvia, November 2017.
- [5] M. Elmonser, H. Ben Chikha, and R. Attia, "Mobile routing algorithm with dynamic clustering for energy large-scale wireless sensor networks," *IET Wireless Sensor Systems*, vol. 10, no. 5, pp. 208–213, 2020.
- [6] L. D. Dongre and V. Gulhane, "Minimizing energy consumption in large scale wireless sensor network using adaptive duty cycle algorithm," in *Proceedings of the International Conference for Convergence for Technology-2014*, pp. 1–5, Pune, India, April 2014.
- [7] S. Liu, J. Pan, L. Kong, and C. Chen, "Time competition flooding in high-density wireless sensor network," in *Proceedings of the 2016 Third International Conference on Computing Measurement Control and Sensor Network (CMCSN)*, pp. 170–173, Matsue, Japan, May 2016.
- [8] L. Wang, J. Mao, L. Fu, H. Zhu, and N. Guo, "An improvement of IEEE 802.15.4 MAC protocol in high-density wireless sensor networks," in *Proceedings of the 2015 IEEE International Conference on Information and Automation*, pp. 1704–1707, Lijiang, China, August 2015.
- [9] V. Sadhu, X. Zhao, and D. Pompili, "Energy-efficient analog sensing for large-scale, high-density persistent wireless monitoring," in *Proceedings of the 2017 13th Annual Conference on Wireless On-Demand Network Systems and Services (WONS)*, pp. 61–68, Jackson, WY, USA, February 2017.
- [10] J. Zhang and Z. Sun, "Serried forwarder routing (SFR): a query-driven routing algorithm for a wireless sensor network with high density of nodes," in *Proceedings of the 2016 International Conference on Network and Information Systems for Computers (ICNISC)*, pp. 114–119, Wuhan, China, April 2016.
- [11] V. Kopta, D. Barras, and C. C. Enz, "An approximate zero IF FM-UWB receiver for high density wireless sensor networks," *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 2, pp. 374–385, 2017.
- [12] M. Hefnawi, "Large-scale multi-cluster MIMO approach for cognitive radio sensor networks," *IEEE Sensors Journal*, vol. 16, no. 11, pp. 4418–4424, 2016.
- [13] J. Guo and H. Jafarkhani, "Movement-efficient sensor deployment in wireless sensor networks with limited communication range," *IEEE Transactions on Wireless Communications*, vol. 18, no. 7, pp. 3469–3484, 2019.
- [14] A. R. de la Concepcion, R. Stefanelli, and D. Trincherro, "Adaptive wireless sensor networks for high-definition monitoring in sustainable agriculture," in *Proceedings of the 2014 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet)*, pp. 67–69, Newport Beach, CA, USA, January 2014.
- [15] B. Wan and W. Zhang, "The lifetime optimization strategy of linear random wireless sensor networks based on mobile sink," in *Proceedings of the 2014 International Conference on Wireless Communication and Sensor Network*, pp. 258–261, Wuhan, China, December 2014.
- [16] B. Li, W. Wang, Q. Yin, R. Yang, Y. Li, and C. Wang, "A new cooperative transmission metric in wireless sensor networks to minimize energy consumption per unit transmit distance," *IEEE Communications Letters*, vol. 16, no. 5, pp. 626–629, 2012.
- [17] M. Abo-Zahhad, M. Farrag, A. Ali, and O. Amin, "An energy consumption model for wireless sensor networks," in *Proceedings of the 5th International Conference on Energy Aware Computing Systems & Applications*, pp. 1–4, Cairo, Egypt, March 2015.
- [18] M. Benaddy, B. E. Habil, M. E. Ouali, O. E. Meslouhi, and S. Krit, "A mutlipath routing algorithm for wireless sensor networks under distance and energy consumption constraints for reliable data transmission," in *Proceedings of the 2017 International Conference on Engineering & MIS (ICEMIS)*, pp. 1–4, Monastir, Tunisia, May 2017.
- [19] M. Abo-Zahhad, M. Farrag, and A. Ali, "Modeling and minimization of energy consumption in wireless sensor networks," in *Proceedings of the 2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, pp. 697–700, Cairo, Egypt, December 2015.
- [20] D. R. Edla, A. Lipare, R. Cheruku, and V. Kuppili, "An efficient load balancing of gateways using improved shuffled frog leaping algorithm and novel fitness function for WSNs," *IEEE Sensors Journal*, vol. 17, no. 20, pp. 6724–6733, 2017.
- [21] M. Djamila and H. Saad, "QGAC: quantum genetic based-clustering algorithm for WSNs," in *Proceedings of the 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 82–88, Limassol, Cyprus, June 2018.
- [22] A. Singh, S. Rathkanthiwar, and S. Kakde, "Energy efficient routing of WSN using particle swarm optimization and V-LEACH protocol," in *Proceedings of the 2016 International Conference on Communication and Signal Processing (ICCSP)*, pp. 2078–2082, Melmaruvathur, India, April 2016.
- [23] K. Li, X. Deng, X. Zhou, and W. Li, "On anycast routing based on parallel evolutionary algorithm," in *Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1691–1695, Sendai, Japan, May 2015.
- [24] U. Cekmez and O. K. Sahingoz, "Parallel solution of large scale traveling salesman problems by using clustering and evolutionary algorithms," in *Proceedings of the 2016 24th Signal Processing and Communication Application Conference (SIU)*, pp. 2165–2168, Zonguldak, Turkey, 2016.
- [25] J. Wang, C. Ju, Y. Gao, A. K. Sangaiah, and G. Kim, "A PSO based energy efficient coverage control algorithm for wireless

- sensor networks,” *Computers Materials & Continua*, vol. 56, no. 3, pp. 433–446, 2018.
- [26] J. Wang, Y. Gao, X. Yin, F. Li, and H. J. Kim, “An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks,” *Wireless Communications & Mobile Computing*, vol. 20189 pages, Article ID 9472075, 2018.
- [27] J. Wang, X. J. Gu, W. Liu, A. K. Sangaiah, and H. J. Kim, “An empower hamilton loop based data collection algorithm with mobile agent for WSNs,” *Human-centric Computing and Information Sciences*, vol. 918 pages, 2019.
- [28] J. Wang, Y. Gao, C. Zhou, R. Simon Sherratt, and L. Wang, “Optimal coverage multi-path scheduling scheme with multiple mobile sinks for WSNs,” *Computers, Materials & Continua*, vol. 62, no. 2, pp. 695–711, 2020.
- [29] C. Duan, J. Feng, H. Chang, J. Pan, and L. Duan, “Research on sensor network coverage enhancement based on non-cooperative games,” *Computers, Materials & Continua*, vol. 60, no. 3, pp. 989–1002, 2019.
- [30] J. Wang, Y. Gao, W. Liu, W. Wu and Se-Jung Lim, and S. Lim, “An asynchronous clustering and mobile data gathering schema based on timer mechanism in wireless sensor networks,” *Computers, Materials & Continua*, vol. 58, no. 3, pp. 711–725, 2019.
- [31] D. Gao, S. Zhang, F. Zhang, X. Fan, and J. Zhang, “Maximum data generation rate routing protocol based on data flow controlling technology for rechargeable wireless sensor networks,” *Computers, Materials & Continua*, vol. 59, no. 2, pp. 649–667, 2019.
- [32] L. Boquete, J. M. Rodríguez-Ascariz, R. Barea, J. Cantos, J. M. Miguel-Jiménez, and S. Ortega, “Data acquisition, analysis and transmission platform for a pay-as-you-drive system,” *Sensors*, vol. 10, no. 6, pp. 5395–5408, 2010.
- [33] N. Thepvilajanapong, T. Ono, and Y. Tobe, “A deployment of fine-grained sensor network and empirical analysis of urban temperature,” *Sensors*, vol. 10, no. 3, pp. 2217–2241, 2010.
- [34] A. K. Sangaiah, D. V. Medhane, T. Han, M. S. Hossain, and G. Muhammad, “Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4189–4196, 2019.
- [35] A. K. Sangaiah, D. V. Medhane, G.-B. Bian, A. Ghoneim, M. Alrashoud, and M. S. Hossain, “Energy-aware green adversary model for cyberphysical security in industrial system,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3322–3329, 2020.
- [36] A. K. Sangaiah, M. Sadeghilalimi, A. A. R. Hosseinabadi, and W. Zhang, “Energy consumption in point-coverage wireless sensor networks via bat algorithm,” *IEEE Access*, vol. 7, pp. 180258–180269, 2019.
- [37] A. K. Sangaiah, A. A. R. Hosseinabadi, M. B. Shareh, S. Y. B. Rad, A. Zolfagharian, and N. Chilamkurti, “IoT resource allocation and optimization based on heuristic algorithm,” *Sensors*, vol. 20, 2020.
- [38] D. Dasgupta, D. Becerra, A. Banceanu, F. Nino, and J. Simien, “A parallel framework for multi-objective evolutionary optimization,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1–8, Barcelona, Spain, July 2010.
- [39] Z.-G. Sun, Zi-W. Zheng, S.-H. Chen, and S.-J. Xu, “An energy-effective clustering algorithm for multilevel energy heterogeneous wireless sensor networks,” in *Proceedings of the 2010 2nd International Conference on Advanced Computer Control*, pp. 168–172, Shenyang, China, March 2010.
- [40] X. Yi and X. Yong-Qiang, “Energy-efficient distributed clustering algorithm based on coverage,” in *Proceedings of the 2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, pp. 32–35, Hong Kong, China, August 2010.
- [41] R. Zhou, M. Chen, G. Feng, H. Liu, and S. He, “Genetic clustering route algorithm in WSN,” in *Proceedings of the 2010 Sixth International Conference on Natural Computation*, pp. 4023–4026, Yantai, China, August 2010.
- [42] M. Dhami, V. Garg, and N. S. Randhawa, “Enhanced lifetime with Less energy consumption in WSN using genetic algorithm based approach,” in *Proceedings of the 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 865–870, Vancouver, BC, Canada, November 2018.
- [43] V. Rajasekar, K. Sathya, and J. Premalatha, “Energy efficient cluster formation in wireless sensor networks based on multi objective bat algorithm,” in *Proceedings of the 2018 International Conference on Intelligent Computing and Communication for Smart World (I2C2SW)*, pp. 116–120, Erode, India, December 2018.
- [44] H. Zhihui, “Research on WSN routing algorithm based on energy efficiency,” in *Proceedings of the 2015 Sixth International Conference on Intelligent Systems Design and Engineering Applications (ISDEA)*, pp. 696–699, Guiyang, China, August 2015.
- [45] T. Ma and W. Shen, “Research on a Hybrid K-Means Clustering Algorithm Based on Improved Genetic Algorithm,” in *Proceedings of the 2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC)*, pp. 502–507, Dalian, China, December 2017.
- [46] R. Chouhan and A. Purohit, “An approach for document clustering using PSO and K-means algorithm,” in *Proceedings of the 2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pp. 1380–1384, Coimbatore, India, January 2018.
- [47] J. Dong and M. Qi, “A New clustering algorithm based on PSO with the jumping mechanism of SA,” in *Proceedings of the 2009 Third International Symposium on Intelligent Information Technology Application*, pp. 61–64, Nanchang, China, November 2009.
- [48] C. Tsai, C. Kang, and M. Chiang, “A quantum-inspired evolutionary algorithm based clustering method for wireless sensor networks,” in *Proceedings of the 2015 Seventh International Conference on Ubiquitous and Future Networks*, pp. 103–108, Sapporo, Japan, July 2015.
- [49] S. Bhushan, R. Pal, and S. G. Antoshchuk, “Energy efficient clustering protocol for heterogeneous wireless sensor network: a hybrid approach using GA and K-means,” in *Proceedings of the 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*, pp. 381–385, Lviv, Ukraine, August 2018.
- [50] C. Jiang, Y. Ren, Y. Zhou, and H. Zhang, “Low-energy consumption uneven clustering routing protocol for wireless sensor networks,” in *Proceedings of the 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, pp. 187–190, Hangzhou, China, August 2016.
- [51] Y. Lu, J. Zhou, and M. Xu, “Biologically inspired low energy clustering method for large scale wireless sensor networks,” in *Proceedings of the 2019 IEEE International Conference of Intelligent Applied Systems on Engineering (ICIASE)*, pp. 20–23, Fuzhou, China, 2019.

- [52] C. Dai and X. Lei, "A novel evolutionary algorithm based on decomposition and adaptive weight adjustment for synthesis gas production." in *Proceedings of the 2015 11th International Conference on Computational Intelligence and Security (CIS)*, pp. 270–273, Shenzhen, China, December 2015.
- [53] C. L. P. Chen, T. Zhang, and S. C. Tam, "A novel evolutionary algorithm solving optimization problems," in *Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 557–561, San Diego, CA, USA, October 2014.
- [54] J. Xiu, Q. He, Z. Yang, and C. Liu, "Research on a multi-objective constrained optimization evolutionary algorithm," in *Proceedings of the 2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pp. 282–286, Beijing, China, August 2016.
- [55] M. Elhoseny, A. Tharwat, A. Farouk, and A. E. Hassanien, "K-coverage model based on genetic algorithm to extend WSN lifetime," *IEEE Sensors Letters*, vol. 1, no. 4, pp. 1–4, Article ID 7500404, 2017.

Research Article

ICS Software Trust Measurement Method Based on Dynamic Length Trust Chain

Wenli Shang ¹ and Xiangyu Xing ^{2,3}

¹*School of Electronic and Communication Engineering, Guangzhou University, Guangzhou 510006, China*

²*Industrial Control Network and Systems Department, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China*

³*Information and Control Engineering Faculty, Shenyang Jianzhu University, Shenyang 110168, China*

Correspondence should be addressed to Wenli Shang; shangwl@gzhu.edu.cn

Received 15 October 2020; Revised 17 February 2021; Accepted 30 March 2021; Published 27 April 2021

Academic Editor: Ting Yang

Copyright © 2021 Wenli Shang and Xiangyu Xing. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at the real-time requirements for industrial control systems, we proposed a corresponding trust chain method for industrial control system application software and a component analysis method based on security sensitivity weights. A dynamic length trust chain structure is also proposed in this paper. Based on this, the industrial control system software integrity measurement method is constructed. Aimed at the validity of the model, a simulation attack experiment was performed, and the performance of the model was repeated from multiple perspectives to verify the performance of the method. Experiments show that this method can effectively meet the integrity measurement under the condition of high real-time performance, protect the integrity of files, and improve the software credibility of industrial control system.

1. Introduction

Industrial control system security, as an important part of the industrial control system, profoundly affects the development of industrial control network-related industries and has a strong degree of industrial relevance and industrial penetration. Information security of industrial control systems has become an important part of the integration of the two industries. The security risks it brings are no longer just “small” problems such as information leakage and unavailability of information systems. With the attack technology and means of industrial control system becoming more and more advanced, complex, and mature, the security threat of industrial control system is becoming more and more serious. Therefore, security measures are urgently needed to deal with the security threat of industrial control system. Trusted computing technology is widely concerned by industrial control industry because it can provide security immunity. As an active defense method, trusted computing

can improve security of industrial control system from inside [1].

Considerable achievements have been made in the credibility of industrial control systems, but most of these achievements are based on traditional computer systems [2, 3]. Because the reliability and real-time requirements of industrial control system are not considered, these methods cannot be directly applied to industrial control system.

For the top-level design of industrial control system security, Cheng et al. [4] proposed a trusted computing-based industrial control security solution. This solution improves the security of industrial control systems through the cooperation of firewall technology, intrusion detection technology, and trusted computing technology. An et al. [5] realized the application of trusted computing technology in power system, realized higher level security protection of power system, set a precedent, and provided a case worthy of reference for the construction of security immunity engineering in other industries.

In trusted computing field, the traditional construction mode of trust chain is only applicable to a single system. For the dual redundancy system, it causes break of transitive trust. Wang et al. [6] proposed a trust chain structure based on a dual redundant system to implement credibility determination in the automatic switching process when an industrial control computer fails. Shang et al. [7–9] discussed the credibility of traditional PLC (programmable logic controller) in industrial control from different angles and achieved specific analysis of common equipment in industrial control systems.

The TCG trust chain has great potential in the construction of a trusted operating system [10], but for the measurement process from the operating system to the application, the structure of TCG trust chain is too simple to meet the diverse requirements of the application layer, and there are fewer applications in the application layer.

IMA (integrity metric architecture), as the first integrity measurement architecture based on TCG standard, is of great significance [11]. IMA determines the trustworthiness of software by performing integrity metric before the program runs, but it does not mean the trustworthiness of the program at run time. Shi et al. [12] proposed BIND (binding instructions and data), a fine-grained attestation service for securing distributed systems, which achieves small granularity dynamic verification by measuring the integrity of key code segments. However, it needs to identify key code segments and establish binding relationship between input and output data before running, which is a complex process. Shankar et al. [13] provided a largely automated system for verifying Clark-Wilson interprocess information-flow integrity. They defined a weaker version of Clark-Wilson integrity, called CW-Lite, which has the same interprocess information-flow guarantees, but which requires less filtering, only small changes to existing applications. But the formal validation capability of the model is not sufficient to formally validate the system.

Li et al. [14] proposed a dynamic trusted application model for privilege isolation, maintaining flexibility in application loading and improving the trustworthiness of the application layer. Garfinkel et al. [15] enabled the measurement of applications by using DRM (digital rights management) approach to deliver trust. Zhang et al. [16] proposed a trustworthiness analysis method for the application layer, which uses a nondisruptive behavioural approach to achieve real-time application metrics.

Industrial control system is a typical system that does not need to be restarted frequently, which makes the application of TCG trust chain in industrial control system very difficult and cannot meet the actual application requirements [17–19]. The purpose of building a trusted operating system is to ensure that the application has a trusted execution environment and that the trust relationship can continue. For industrial control systems, based on meeting the reliability and real-time requirements of industrial control systems, improving the credibility of application programs is a very critical issue [20, 21]. In this paper, we start with the static measurement method, focus on real-time, and build a trusted approach at the application layer. Compared with the

existing literature, this paper has the following major contributions:

- (1) A software component analysis method is proposed that takes security sensitivities into account. This approach enables a hierarchical protection scheme for different files by dividing the software components by considering the security sensitivity of the files. The solution can effectively improve the real-time performance of applications when performing integrity checks.
- (2) A new dynamic length chain of trust transfer model is proposed. The model can be adapted to different real-time requirements and complete integrity verification on the basis of meeting the real-time requirements of industrial control systems. This new dynamic length chain of trust structure has the features of easy updating, dynamic structure, and adjustable real-time requirements.
- (3) A software trustworthiness verification model based on dynamic length chains of trust is proposed. Adopt the concept of differentiated design of the application software's own components and the system's common components to achieve classification and management of different components and to optimise the efficiency of integrity verification. Effectively improve the efficiency of system integrity verification, reduce the burden of trusted computing on the system so that trusted computing technology means can continue to be used in the absence of system resources and can be effectively compatible with the old industrial control system, and enhance the system's trustworthiness as much as possible at a lower upgrade cost. The update process of the model is also discussed, which is characterised by easy software updates.

The rest of this paper is organized as follows. ICS software analysis is discussed in Section 2. A software trust measurement model based on dynamic length trust chain (DLTC) is proposed in Sections 3. The effectiveness of the algorithm is verified by a simulation experiment in Section 4. Finally, the conclusions are given in Section 5.

2. ICS Software Analysis

2.1. Real-Time. Industrial control system is facing a large number of new security challenges, and industrial control computer is undertaking more and more information security responsibilities. Information security protection methods generally adopt the combination of active defense and passive defense. As an important technology of active defense, trusted computing is becoming more and more important. Generally, the combination of dynamic and static methods is used to ensure the credibility of the system. Integrity measurement is a common static measurement method. However, it also has an inherent disadvantage: integrity measurement of trusted computing is usually performed before the program runs. For services with real-

time requirements, careful consideration must be given to whether to use static measurement methods for integrity measurement [22].

Real-time means that the input, calculation, and output of the signal are completed in a very short time and processed in time according to the changes in the generation process. Real-time is the ability to perform specified functions within a limited time and respond to external asynchronous events. Emphasis is on the specified time, as long as the completion within the specified time is real-time [23].

Specifically, for any stimulus-response system, there is a time from the stimulus input to the response output, that is, the stimulus-response period T , which represents the time response capability of the system.

If the response time T of the system can meet the requirement of the response time t specified by the system, that is, $t \leq T$, the system is a real-time system.

In industrial control systems, two main factors are affecting the real-time performance of the system: on the one hand, the real-time nature of the unit components in the system. That is, controllers, sensors, and actuators must meet real-time requirements. On the other hand, it refers to the real-time nature of the industrial communication network, and the information interaction between field devices must be completed within a certain time.

According to the requirements of different systems for real-time requirements, field information can be divided into real-time information and nonreal-time information. Real-time information must be processed on time, with high requirements for real-time performance. Priority must be given to prevent system failure. At the same time, the credibility of this operation is higher. The execution of this part of the program is related to the credibility of the system's fault handling behavior.

The integrity measurement method usually increases the delay and affects the real-time performance of the system. Therefore, for systems with high real-time requirements, integrity is to measure integrity on the premise of meeting real-time requirements.

Integrity measurement and real-time are usually mutually limited. For hard real-time systems, the delay fluctuation in integrity measurement may have a significant impact, resulting in that system cannot operate normally. For soft real-time system, the impact of integrity measurement is less than that of hard real-time system. At the same time, the availability of industrial control system is very important. There is a strong positive correlation between real-time and availability. Generally speaking, availability and real-time should be considered first when considering the security of industrial control system.

2.2. File Type. Documents have different security sensitivities and different levels of security protection. For security-sensitive files, more security measures need to be added to ensure that they are secure. For security-sensitive files, such as dynamic link library, static link library, and Perl script, this paper analyzes the file characteristics of Linux platform and the emphasis of integrity measurement.

This paper mainly analyzes and studies the common file types in Linux system. For example, the binary file is the traditional executable file, while the script language program is generally described as a text file.

This paper considers that security-sensitive files under the Linux platform which can be divided into the following three categories: the first category is executable files, which includes the files that can be executed with executable permission, and the files that can be executed after getting executable permission but have no executable permission at present. These files include the main files that can be directly run, including dynamic link libraries, static link libraries, binary files, and scripts. The second category is the files containing sensitive data. These files store information that may be used during program operation, such as files that record the hash value of files. They need to be verified for integrity to prevent hackers from modifying it after obtaining administrator rights of these files. The third category is the user-defined security-sensitive files. Users can define files as security-sensitive files according to actual needs.

2.3. Component Analysis Method Based on Security Sensitivity. The previous paper analyses the software requirements and characteristics of industrial control systems from three perspectives. To better describe the internal connections and characteristics of these files, a two-dimensional attribute is constructed to describe the characteristics of the files. Through cluster analysis of the file characteristics, the internal connections of the files are obtained, and the files are decomposed to meet the needs of establishing a real-time trust chain structure.

Software attributes are constructed from two dimensions: the first is the weight of security sensitivity. Different levels of security sensitive files need different weights, which need to be determined artificially and protected by different protection levels. This is a "hierarchical protection" method [24], which can effectively deal with complex practical situations.

The second is the relationship and function between calling and called. The calling relationship in a program is usually organized by function. The granularity of analyzing software behavior based on function is too small, which leads to too many details involved in measurement. Usually, the number of files involved in a program that performs a specific function is fixed, and not all files are involved. By analyzing the calling relationship of a specific function, the number of files that need to be verified when using the function can be effectively reduced, so as to improve the real-time performance of integrity measurement.

Due to the difference of software structure, it is difficult to analyze the calling and called functions with automatic method, and it depends more on experience. Taking OpenPLC as an example, it is mainly based on the inclusion of header files, and the calling relationship of program files is determined by the operation of header files. OpenPLC file structure is clear and easy to identify. Usually, each folder is a file that performs independent functions.

Through the above two attributes, different files can be classified and processed. The software is written with different software structure, and the analysis is slightly different. A well-designed software should meet the requirements of “high cohesion and low coupling.” For such software, it is easy to handle at the file and folder level.

According to these two principles, files with similar functions are put in the same “package.” The package consists of several files with similar functions and security sensitivity [25]. These files should show the relationship between the calling and the called. The concept of package essentially defines the delay of component, but the granularity of the component is different from the traditional definition [26, 27]. This method puts more emphasis on the security sensitivity of files, which is an improvement of component definition.

2.4. Dynamic Length Trust Chain. In the industrial control system, the software update is slow, and it takes a lot of time to realize the dynamic measurement method in the industrial control system [28]. At present, static method is widely used because of its simple and easy deployment [29].

Generally speaking, these methods are easy to implement and highly customized, and traditional models use TPM as the source of trust. As a trusted root, TPM has been proved to be a feasible solution and has been widely used. Many scholars put forward the construction technology of virtual trusted root in cloud environment [30–32]. For some industrial control systems without TPM, the trusted root based on USB can be used, which can also bring the expected effect [33, 34].

The trust chain length of TCG organization is fixed, and the measurement time is relatively fixed, which cannot meet the real-time requirements of industrial control system. If we give up the measurement of some documents directly, it is difficult to guarantee their credibility. Therefore, we need a dynamic length of trust chain, and cut the length of the trust chain to meet the requirements of trust degree. At the same time, the chain trust transfer model cannot effectively describe the call and transfer of control between applications. Even for entities with measurement capability, with the increase of the number of entities in the computing platform, the chain trust transfer model will become difficult to manage.

In view of these shortcomings, this paper proposes a Dynamic length trust chain (DLTC) structure for real-time industrial control systems. This structure effectively solves the problems of fixed length of trust chain, fixed measurement time, and poor real-time performance. At the same time, this structure can also effectively describe the calling relationship between software. The description of software dependency can adapt to the current software environment.

The file package sequence obtained by using the analysis method based on security sensitivity and components is $F_0, F_1, F_2, \dots, F_n$, recorded as Ω , as shown in (1). F_0 is the main function; n is the number of file package; there are $n+1$ file packages in Ω :

$$\Omega = \{F_0, F_1, F_2, \dots, F_n\}. \quad (1)$$

For a program, the description of its package is shown in (1), and then a corresponding chain of trust can be described as an ordered sequence as described in (2); $\varphi_i = 1$ means the file package is in TL; $\varphi_i = 0$ means the file package is not in TL; in particular, $\varphi_0 = 1$ means the trust chain always contains the main function.

$$TL = \{\varphi_0, \varphi_1, \varphi_2, \dots, \varphi_n\}. \quad (2)$$

The calculation method of the length definition of the trust chain is shown in

$$\text{len} = \sum_{i=0}^n \varphi_i. \quad (3)$$

Equations (1) and (2) describe the elements contained in the trust chain. In essence, the trust chain is still a chain structure trust chain, but because the successor nodes of each node in the trust chain are dynamically determined, it forms a tree structure trust chain.

3. Software Trust Measurement Model Based on DLTC

3.1. Real-Time Mathematical Description. For an entity, the system response time requirement is T , which means that the response time [35] meets the following equation:

$$t \leq T. \quad (4)$$

For an entity, the response time without adding the integrity measurement method is t_1 , and the time required for its integrity measurement is t_2 , and then the total response time satisfies the following equation:

$$t = t_1 + t_2. \quad (5)$$

Then, the expected maximum response time of the chain of trust is defined, which is as shown in equation (6). In equation (6), α is the dynamic coefficient, and $0 \leq \alpha \leq 1$. Its existence is to leave a margin for the estimation error of the system in the required measurement time, so as to ensure that the system will not fail due to exceeding the response time requirement under hard real-time conditions.

$$T_{h \max} = \alpha \times (T - t_1). \quad (6)$$

For any package F_i ($1 \leq i \leq n$), the files in it are marked as f_{ij} . There are two attributes for anyone, one is the file size, marked as s_{ij} ; the other is the expected measurement time of the file, marked as τ_{ij} , which describes the end of the comparison of the completeness of the hash calculation value integrity check result of the file.

The expected measurement time τ_{ij} for any file satisfies the following equation:

$$\tau_{ij} = a + b \times s_{ij}. \quad (7)$$

The calculation of equation (7) is obtained by least square fitting, which describes a relationship. The time

required for a file to perform integrity measurement consists of two parts. One is the file I/O time, and the other is the time required for hash calculation. In equation (7), a describes the fixed overhead time of file I/O, and b describes the measurement time that increases as the file size increases. This part of the time is mainly generated by the hash operation.

The actual values of the parameters a and b need to be calculated according to the actual configuration of the system. The calculation process of this parameter in the experimental environment of this paper is detailed in Section 4.2.

Then, the integrity measurement time of each file package is $\sum_j \tau_{ij}$, which is defined as the measurement time sum of all files contained in the file package.

The most time-consuming step in the dynamic generation of a complete trust chain is the hash operation, but other smooth processing also consumes a lot of time. The main steps of trusted chain generation include the following parts: (1) Due to the serial original in the TPM, the backlog of unfinished operations will affect the subsequent operations, which will cause delays, which is recorded as t_{TPM} , as shown in Figure 1. (2) Processing hash metric list (HML) operations requires time. HML is defined in Section 3.2. This part of the time can be divided into HML reading time, recorded as t_{HMLr} , and time to write HML, recorded as t_{HMLw} . (3) The time consumed to generate the random number, process the random number, and generate the chain of trust is t_{randm} . (4) The time to calculate the integrity of all the files in the chain of trust.

$T_{h \max}$ can also be described in the following equation:

$$T_{h \max} = t_{\text{TPM}} + t_{\text{HMLr}} + t_{\text{randm}} + \sum_{i=0}^n \text{TL}(i) \times t_{Fi}. \quad (8)$$

3.2. Hash Metric List. For a program, its related information is recorded in a text file, called a hash metric record table, abbreviated as Hash Metric List (HML). It is stored in a trusted storage space controlled by the TPM, it is encrypted by the encryption algorithm contained in the TPM, and the key is stored in the TPM to ensure the security of the HML. The file structure of the HML is shown in Figure 2.

The main information stored in HML includes the following contents: the file structure of the software and all files, the number of measurements transferred by each file, total number of measurements, the file size, and the integrity measurement results of each file.

In order to prevent the HML file from being tampered, integrity measurement and report should be carried out before using the HML file every time to determine the credibility of the HML file.

To maintain HML more conveniently and efficiently, we divide HML file into two types, the first one is a system hash metric record table, denoted as SHML (system hash metric list), and the second one is an application hash metric record table, denoted as AHML (application hash metric list). SHML is used to store the hash value measurement results of public resources in the system, and AHML is used to record the hash value measurement results of the application's files.

The main reasons for adopting this design method are as follows:

- (1) Storing all records in a unified HML will lead to too large HML file and low search efficiency, which cannot meet the design goal. The use of large HML files will have a great impact on the real-time performance of the system.
- (2) The component-based design method produces a large number of public resources. Many of them are public resources in the system as dynamic link library (DLL). A large number of services call DLL. If each HML stores relevant information, it will cause a great waste of resources.
- (3) Credibility of integrity measurement results can be considered as short-term rather than long-term, or invalid after measurement. Whether two integrity measures are needed in two calls to common resources in a short time is worth discussing. By considering the validity period of integrity measurement, we can effectively reduce the time of integrity measurement and improve the real-time performance of the system.
- (4) Consider the security principles of "least privilege" and "as needed." These DLL modules only serve specific programs. If they are all stored in a unified HML file, it may bring information security risks.

In summary, the HML is divided into a system-maintained SHML for controlling common components and an application-specific AHML.

As described in Section 2.3, through the analysis of software components, the files in a complete software are classified, and the loose file structure is changed into a compact file package structure. The granularity of the package can be adjusted according to the measurement requirements. AHML is dynamically formed in the trust chain constructed in this paper. To ensure information security, two files are used to manage AHML, AHML-H, and AHML-F.

AHML-H is used to store software packages, absolute path names and integrity metrics. It stores sensitive information. In order to prevent tampering, it separates the information that needs to be modified from the information that does not need to be modified during the operation. AHML-H stores the integrity measurement results and encrypts them with the encryption function provided by TPM.

AHML-F is used to store packages. The main information it stores is the file package, package size, measurement times, measurement interval, security sensitivity weight, and total measurement times. The file is dynamic when the trust chain is generated. In order not to store sensitive check value information and ensure security, the random number selection algorithm is used to achieve the measurement times, so as to achieve better robustness and ensure the correct operation of the program when the measurement times have problems.

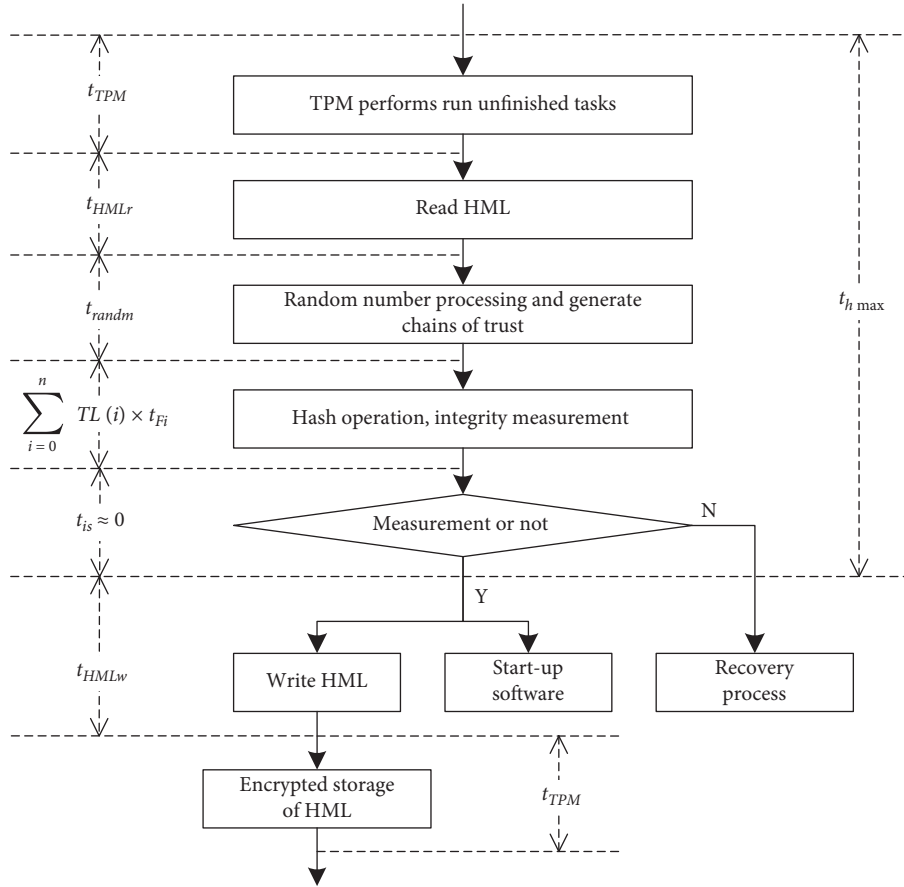


FIGURE 1: Trust chain generation process and time consumption.

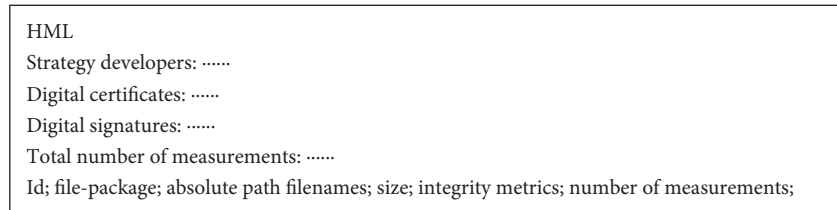


FIGURE 2: File structure of the HML.

Implementing AHML as two files has the following advantages: firstly, the separation of sensitive information and nonsensitive information is realized. AHML-H and AHML-F are stored in trusted storage areas. There is no sensitive information in AHML-F, and it will be modified dynamically during the running of the program to ensure its security. Once the memory leak occurs, the impact will be very small. AHML-H file stores sensitive integrity measurement results, which can ensure its security by only reading and not modifying.

Secondly, unnecessary information can be hidden. AHML-H file itself does not reflect those files that need to be measured, which can reduce the time of file path and integrity measurement results appearing in memory and prevent the information security problem that the software cannot be loaded into memory when using static measurement method.

Finally, the size of related files can be reduced, and the complexity of AHML-H file retrieval can be reduced. It can save time to measure while retrieving. By separating reading AHML-H file from writing AHML-H file, the writing operation of AHML file is independent of measurement, which can save measurement time.

3.3. Trust Chain Generation Method Based on Roulette Rules. For a software, a series of packages are obtained by using the “analysis method based on security sensitivity and component,” which maintain an AHML-F file and an AHML-H file. The operating system maintains a SHML file. For AHML-F, the data item “measures” constitutes an array M of lengths $1 \times n$, where M_i is number of times the i -th file package was measured. The combination of arrays M and

random numbers determines the generation of trust chains. The random number is generated by the TPM. The random number generator is one of the several functions provided by the TPM. Its data item “security sensitivity weight” constitutes an array W of lengths $1 \times n$, where W_i is the weight of the i -th file package.

Each time TPM generates a random number R , the mapping between the random number and the file package is generated by the roulette selection method. The steps of generating trust chain file by roulette selection method are as follows.

3.3.1. Handling Arrays M and Arrays W . The array W exists in the form of characters and needs to be converted into numeric values. The array W reflects the adjustment of the trust chain structure according to the security sensitivity of the package, which affects the structure of the trust chain in a proportional way. Depending on the sensitivity, the values are 4, 3, 1, and 0.5.

To improve the robustness of the algorithm, restrictions are added. For packages that meet the conditions shown in equation (9), in the selection of trust chain, the probability of being selected is 1, where $\max(-)$ means calculating the maximum value of the array. k is the limited number of times, which can be adjusted according to the actual situation. In this paper, $k = 5$.

$$\max(M) - M_i > k. \quad (9)$$

Equation (9) is to ensure that the difference between the number of times any two packages are measured is not greater than 5, so as to ensure that in extreme cases, non-security sensitive data can also get a limited number of integrity measures. The robustness of the algorithm is improved.

3.3.2. Selection Probability. Roulette selection method, also known as proportional selection method, is based on the proportion of individuals in the whole. First, the array M is updated by the following equation:

$$M_i = \max(M) - M_i + 1. \quad (10)$$

Then, the probability of selection is calculated by the following equation:

$$P(F_i) = \frac{M_i \times W_i}{\sum_{i=1}^n M_i \times W_i}. \quad (11)$$

3.3.3. Cumulative Probability. Selection probability of each package is calculated by the following equation:

$$q_i = \sum_{j=1}^i P(F_j). \quad (12)$$

3.3.4. Random Number Selection. The random number R generated by the TPM is used to determine the file package

to be selected. If $R < q$ [1], select package 1; if $q[h-1] < R < q$ [h], select the h -th individual.

3.3.5. Repeating the Above Process until Sufficient Data Are Generated. In this way, the trust chain structure can be generated dynamically. Due to the robustness of the algorithm, the measurement times of the algorithm will not be greatly different.

3.4. Trust Chain Update Mechanism. The trust chain of TCG chain structure adopts the extension operation of hash, which makes it difficult to update. Once the hash value of a node is updated, the trust chain needs to recalculate the hash value and carry out the extension operation, which leads to a lot of calculation work. To solve this problem, the star trust chain stores hash values for different nodes, which can effectively solve the problem of updating. The trust chain structure proposed in this paper has the characteristics of star structure trust chain and is easy to update.

The update process of file modification is as follows:

- (1) Update AMHL-H file. Update the files that need to be updated and modify the corresponding records in the AMHL-H file.
- (2) Update the AHML-F file. Update the file size of the corresponding package.

The update process of file addition is as follows:

- (1) Calculate the distance between the feature description of the new files and the cluster center; classify the new files to the closest file package.
- (2) Update AMHL-H file. Add new records to the AMHL-H file.
- (3) Update the AHML-F file. Update the file size of the corresponding package.

The update process of file deletion is as follows:

- (1) Update AMHL-H file. Delete related records located in AMHL-H files.
- (2) Update the AHML-F file. Update the file size of the corresponding package.

4. Experiment

4.1. Simulation Platform Building. The configuration of the experimental platform is shown in Table 1. Due to the limitation of experimental conditions, TPM v1.2 is adopted. Theoretically, all functions implemented on TPM v1.2 can be implemented on TPM 2.x [36].

This experiment only verifies the effectiveness, and the performance experiment only represents the running effect in the current experimental environment. The benchmark data measured in Section 4.2 only represent the current experimental platform.

4.2. Benchmark Data Measurement. In order to objectively measure the impact of device configuration on the time

TABLE 1: Configuration information for the prototype system.

| Name | Version |
|-----------------------------------|---------------------------------|
| System | Ubuntu 18.0 |
| CPU | Core i5-4200 M 2.50 GHz, 4 core |
| Memory | 4 GB |
| Secondary cache | 4 MB |
| TPM version | 1.2 |
| Gun multiple precision arithmetic | 6.1.2 |
| Trousers | 0.3.14 |
| TPM-tools | 1.3.9.1 |
| TPM chip version | 1.2.0.7 |
| TPM spec level | 2 |
| TPM errata revision | 1 |
| TPM vendor ID | ETHZ |
| TPM version | 01010000 |
| TPM manufacturer info | 4554485a |

consumption of TPM hash operation, the following experimental scheme is used to test:

- (1) Randomly generate 1024 files, the file size is from 1 KB to 1024 KB, and the interval is 1 KB.
- (2) Hash these files and extend them to PCR through extend operation, and record the time required for each operation, the unit of measurement is *ms*, and measure 100 times repeatedly.
- (3) Processing data: remove the data less than the smaller quartile, remove the data greater than the larger quartile, and calculate the arithmetic mean of the remaining data to replace the whole data.
- (4) The data calculated in step 3 are fitted by least square method, and the fitting calculation is carried out by

$$T_{\text{hash}} = 0.0022 \times F_s + 0.0621, \quad (13)$$

where T_{hash} is the time required to hash the file, and the unit of measurement is *ms*; F_s is the size of the file to be measured; the unit of measurement is KB.

The determination coefficient of the fitting equation is $r^2 = 0.9989$, which indicates that the fitting result is good.

4.3. Case Analysis Based on OpenPLC. OpenPLC is an easy-to-use programmable logic controller based on open-source protocols [37, 38]. For OpenPLC, the first dimension of its file attributes is analyzed in two steps:

Step 1: filter the installation file name and authority.

Through the LS command in Linux, output all the file names and related authorities, and filter out the files that can be considered as the security-sensitivity weight of D through reverse filtering. Files with such characteristics are usually as follows:

- (1) C/C++ language source code, Java source code, Makefile and other files with source code: script files are not included. For OpenPLC, these files are only

used in the installation, and do not work in the subsequent software operation.

- (2) Config files used during software installation: these files serve Linux software installation. Config file used in the software running process is not included.
- (3) Intermediate files in the compilation process: there are a lot of C/C++ programs in OpenPLC, which will produce some intermediate files in the process of compiling.
- (4) Auxiliary function files: these files are used to assist users and have nothing to do with the trusted operation of the software. These files mainly include help documents, readme, installation logs, and copyright license files.
- (5) Files unrelated to the installation platform: in OpenPLC, there are related files for Windows platform and docker container. Whether these files have security sensitivity depends on different experimental platforms. For the platform used in this experiment, these files are useless.
- (6) When the software has strong robustness, some image files can be considered as not security-sensitive. The important premise is that the software has enough robustness and will not fail because of the lack of these image files.

Step 2: through manual measurement, security sensitivity of files can be determined more accurately.

For OpenPLC, some folders are used to store software output files. These are software outputs and have nothing to do with the operation of the software. The security of these outputs can be guaranteed by data encryption, which is beyond the scope of this article.

According to the above method, the OpenPLC package number and file size are shown in Table 2. As shown in Table 2, the total size of the file package is 1092.5 KB. According to (13), the expected measurement time is 24.0961 *ms*, the maximum file package size is 861.4 KB, and the expected measurement time is 18.5721 *ms*, accounting for 77% of the total expected measurement time. The main reason for this problem is that the file package contains files with large memory consumption.

4.4. Effectiveness Analysis. After successful installation of the experimental environment, comparative experiments are designed to verify the effectiveness of the algorithm. The validation of the algorithm mainly considers whether the algorithm can effectively prevent illegal start-up behavior.

Table 3 shows the validity of the algorithm. The above analysis shows that the algorithm is effective.

4.5. Performance Analysis. Dynamic length trust chain mainly solves the strict real-time requirements of industrial control system, so the fluctuation of its performance is very important for the credibility of industrial control system software and the performance analysis of model. Real-time

TABLE 2: Package number and size.

| No. | Directory | Size | Security sensitivity |
|-----|--------------------------|---------|----------------------|
| 0 | ./start_openplc.sh | 46 | 4 |
| 1 | ./utils/dnp3_src/ | 182300 | 4 |
| 2 | ./utils/libmodbus_src | 40139 | 4 |
| 3 | ./utils/matiec_src | 1161242 | 4 |
| 4 | ./utils/st_optimizer_src | 32136 | 4 |
| 5 | ./webserver/otherfile | 8615563 | 4 |
| 6 | ./webserver/static | 538565 | 1 |
| 7 | ./webserver/core | 396202 | 3 |
| 8 | ./webserver/lib | 213255 | 3 |
| 9 | ./webserver/scripts | 7303 | 3 |

TABLE 3: Algorithm validation results.

| Test case | Use integrity measurement | Do not use integrity measurement |
|--|---------------------------|----------------------------------|
| Use correct start_openplc.sh file | Successful | Successful |
| Modify one of the security-sensitive files | Unsuccessful | Successful |
| Modify the file bool_true.png that does not affect the application | Successful | Successful |
| Update libmodbus.o with security measures | Successful | Successful |
| Infect files with a virus bootkit [39–41] | Unsuccessful | Successful |

analysis can be done from two aspects: low delay requirement and high delay requirement.

First, the test is executed when the real-time requirement is low, all metrics can be executed, and the expected maximum response time is limited. This time limit can fully meet the package distribution shown in Table 2. Under this real-time condition, the experimental data of 100 repeated tests are shown in Figure 3.

In the data shown in Figure 3, the maximum consumption time is 25.6791 ms, the minimum consumption time is 23.0454 ms, the average consumption time is 24.2340 ms, and the variance is 0.3627. From the data fluctuation and variance data shown in Figure 3, we can see that the stability of the algorithm is very good, and the data fluctuation is small. In the process of integrity measurement, the expected maximum response time $T_{h \max}$ will not be exceeded, which can meet the requirements of low real-time.

Secondly, test is performed for a situation that the real-time requirement is high and a large number of measurements cannot be carried out. In this case, the scheduling ability of the algorithm determines the number of times the file is measured. The limit dynamic coefficient is $\alpha = 0.9$, the maximum expected response time is $T_{h \max} = 24$ ms, and the time required for OpenPLC to perform a hash operation is $T_0 = 24.0961$ ms, here $T_0 > T_{h \max}$. At the same time, considering the fluctuation of calculation time in the measurement process, the integrity of all files cannot be calculated in the measurement process. In order to meet the real-time and robust requirements of industrial control system, the performance analysis of the algorithm should be carried out from two aspects: (1) considering the fluctuation of the algorithm. The fluctuation of algorithm will lead to timeout and destroy the availability of industrial control system; (2) considering the fluctuation of packet measurement process. The number of times each software package is measured should be as stable as possible to ensure that each

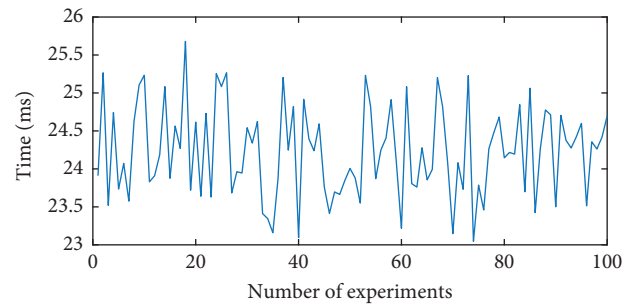
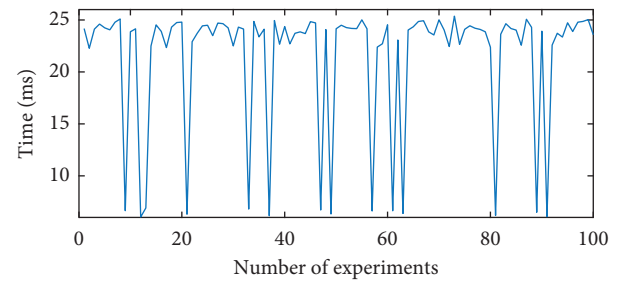


FIGURE 3: Running time during low real-time request.

FIGURE 4: Running time ($T_{h \max} = 24$ ms).

file has the opportunity to be measured, so as to improve the credibility of the whole software system.

In Figure 4, the maximum value is 25.3655, and 54% of them is larger than $T_{h \max}$. All samples are less than $T_{h \max} \cdot \alpha = 0.9$ is used to describe the margin left by the algorithm. It can be seen that the algorithm does not exceed the limit. The results show that the fluctuation of the algorithm is reasonable and will not cause the program execution to fail beyond the time limit.

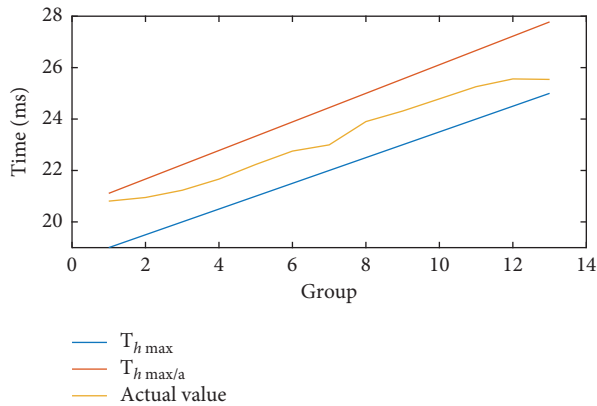


FIGURE 5: Maximum running time.

When dynamic coefficient $\alpha=0.9$ and the maximum distribution of the expected maximum response time $25 \text{ ms} \geq T_{h \max} \geq 19 \text{ ms}$, distribution of the maximum running time is shown in Figure 5.

5. Conclusions

According to the real-time requirements of industrial control system, as well as the operating system and application software rarely involved in TCG related research, the corresponding trust chain construction method is proposed in this paper. According to the characteristics of industrial control system software, a component analysis method based on security sensitivity weight is proposed from three aspects of real-time requirements, component characteristics, and file characteristics. Based on the analysis of traditional trust chain structure, a dynamic length trust chain structure is proposed. The generation process of this trust chain structure is described in detail, and an example is analyzed.

The effectiveness of the model is evaluated from two aspects. The effectiveness of the model is verified by simulation attack experiments, and the effectiveness of dynamic length trust chain is analyzed by simulating the impact of different attack behaviors on file changes. Experiments show that this method can effectively deal with various attacks, protect the integrity of the file, and improve the credibility of the program. The performance of the model is analyzed by repeated experiments. Performance analysis experiments show that the method can meet different real-time requirements.

This paper studies the measurement method of operating system and application in trust chain and proposes a new trust chain structure. However, this method still needs to be further improved. At present, the degree of automation of the analysis process is low, which requires the intervention of human experience, and the process is more complex. The real-time condition is limited, because information security technology will inevitably lead to delay, and the harsh real-time requirements cannot be met. These are our further research directions.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the “National Key R&D Program of China” (2018YFB2004200), the Open Project of Zhejiang Lab “Construction Technology of Local High Security Trusted Execution Environment for Edge Intelligent Controller” (2021KF0AB06), and the National Natural Science Foundation of China “Research on anomaly detection and security awareness method for industrial communication behaviours” (61773368). The authors would also like to acknowledge the helpful comments and suggestions of the Industry Control System Security Software Group. Their efforts are greatly appreciated.





References

- [1] R. R. Schell and M. F. Thompson, “Platform security: what is lacking?” *Information Security Technical Report*, vol. 5, no. 1, pp. 26–41, 2000.
- [2] T. Morris, R. Vaughn, and Y. Dandass, “A retrofit network intrusion detection system for MODBUS RTU and ASCII industrial control systems,” in *Proceedings of the 2012 45th Hawaii International Conference On System Sciences*, pp. 2338–2345, IEEE, Maui, HI, USA, January 2012.
- [3] S. L. P. Yasakethu and J. Jiang, “Intrusion Detection via Machine Learning for SCADA System Protection,” in *Proceedings of the 1st International Symposium For ICS & SCADA Cyber Security Research*, pp. 101–105, Guildford, Leicester, UK, September 2013.
- [4] S. Cheng and Z. Lianggao, “An information security solution scheme of industrial control system based on trusted computing,” *Information And Control, China*, vol. 44, no. 5, pp. 628–640, 2015, in Chinese.
- [5] N. Y. An, Z. H. Wang, and B. H. Zhao, “Research and application of trusted computing in electric power system,” *Journal of Information Security Research*, vol. 3, no. 4, pp. 353–358, 2017, in Chinese.
- [6] L. Wang, M. H. Yang, Z. L. Liu, and J. Q. Zheng, “Trust chain generating and updating algorithm for dual redundancy system,” *Journal on Communications*, vol. 38, no. 1, pp. 1–8, 2017, in Chinese.
- [7] W. L. Shang, X. Y. Xing, X. D. Liu, L. Yin, and E. Y. Gao, “Research on security enhancement of total shipcomputing environment based on trusted computing,” *Ship Science and Technology*, vol. 42, no. 13, pp. 125–129, 2020, in Chinese.
- [8] W. L. Shang, L. Yin, X. D. Liu, and J. M. Zhao, “Construction technology and application of industrial control system security and trusted environment,” *Netinfo Security*, vol. 6, pp. 1–10, 2019, in Chinese.
- [9] M. J. Li, L. D. Wang, W. Xiong, and H. Ding, “Research on trusted computing constructing technology for PLC system,” *Software Guide*, vol. 16, no. 11, pp. 168–175, 2017, in Chinese.
- [10] C. Li, R. Li, and L. Zhuang, “Formal analysis of trust chain,” in *Proceedings of the Second International Conference on Networks Security*, pp. 111–116, Wireless Communications and Trusted Computing, Madurai, India, July 2010.
- [11] R. Sailer, X. Zhang, T. Jaeger, and L. Van Doorn, “Design and implementation of a TCG-based integrity measurement

- architecture,” in *Proceedings of the USENIX Security Symposium*, pp. 223–238, San Diego, CA, August 2004.
- [12] E. Shi, A. Perrig, and L. Van Doorn, “Bind: a fine-grained attestation service for secure distributed systems,” in *Proceedings of the IEEE Symposium on Security and Privacy (S&P’05)*, pp. 154–168, IEEE, Oakland, CA, USA, July 2005.
- [13] U. Shankar, T. Jaeger, and R. Sailer, “Toward automated information-flow integrity verification for security-critical applications,” in *Proceedings of the Network and Distributed System Security Symposium, NDSS 2006*, San Diego, California, USA, October 2006.
- [14] X. Y. Li and C. X. Shen, “Research to a dynamic application transitive trust model,” *Journal of Huazhong University of Science and Technology (nature Science)*, vol. 33pp. 310–312, z1, 2005, in Chinese.
- [15] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh, “Terra,” *ACM SIGOPS Operating Systems Review in Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, vol. 37, no. 5, pp. 193–206, Bolton Landing, NY USA, October 2003.
- [16] F. Zhang, M. D. Xu, H. C. Chao et al., “Real-time trust measurement of software: behavior trust analysis approach based on noninterference,” *Journal of Software*, vol. 30, no. 8, pp. 2268–2286, 2019, in Chinese.
- [17] X. Y. Li, Z. Han, and C. X. Shen, “Transitive trust and performance analysis in Windows environment,” *Journal of Computer Research and Development*, vol. 44, no. 11, pp. 1889–1895, 2008, in Chinese.
- [18] T. Huang and C. X. Shen, “A trusted bootstrap solution based on a trusted server,” vol. A01, pp. 12–14, Journal of Wuhan University, 2004, in Chinese.
- [19] L. Yan, J. Zhang, and A. Zhang, “Scheme of trusted bootstrap based on general smart card,” *Journal of Beijing University of Technology*, vol. 43, no. 1, pp. 100–107, 2017, in Chinese.
- [20] L. H. Fu and D. Wang, “Research on Trust Evaluation of Secure Bootstrap in Trusted Computing Based on Fuzzy Set Theory,” in *Proceedings of the 2010 International Conference On Machine Learning And Cybernetics*, pp. 592–595, IEEE, Qingdao, China, July 2010.
- [21] K. Balasubramanian and A. M. Abba, *Secure Bootstrapping Using the Trusted Platform Module*, IGI Global, Pennsylvania, USA, 2018.
- [22] L. Davi, A. R. Sadeghi, and M. Winandy, “Dynamic integrity measurement and attestation: towards defense against return-oriented programming attacks,” in *Proceedings of the 2009 ACM Workshop on Scalable Trusted Computing*, pp. 49–54, ACM, Chicago Illinois, USA, November 2009.
- [23] Z. Quan, X. Yuan, and Y. Zhu, “Real-time flow control system based on siemens PLC,” in *Proceedings of the 2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1703–1708, IEEE, Tianjin, China, August 2019.
- [24] C. Weiping, “The application of trusted computing 3.0 in classified protection standard system 2.0,” *Journal of Information Security Research*, vol. 4, no. 7, pp. 633–638, 2018, in Chinese.
- [25] Y. Yu, Y. G. Liu, and J. Gu, “Analysis and measurement of components trust relationship in internetware system,” *Netinfo Security*, vol. 18, no. 1, pp. 31–37, 2018, in Chinese.
- [26] G. J. Holzmann, “Software components,” *IEEE Software*, vol. 35, no. 3, pp. 80–82, 2018.
- [27] B. Wang, Y. Chen, S. Zhang, and H. Wu, “Updating model of software component trustworthiness based on users feedback,” *IEEE Access*, vol. 7, pp. 60199–60205, 2019.
- [28] T. Suzuki, *TPM in Process Industries*, Routledge, UK, London, 2017.
- [29] C. Shen, H. Zhang, H. Wang et al., “Research on trusted computing and its development,” *Science China Information Sciences*, vol. 53, no. 3, pp. 405–433, 2010, in Chinese.
- [30] M. Chiregi and N. J. Navimipour, “Trusted services identification in the cloud environment using the topological metrics,” *Karbala International Journal of Modern Science*, vol. 2, no. 3, pp. 203–210, 2016.
- [31] R. Shaikh and M. Sasikumar, “Trust model for measuring security strength of cloud computing service,” *Procedia Computer Science*, vol. 45, pp. 380–389, 2015.
- [32] M. Chiregi and N. Jafari Navimipour, “Cloud computing and trust evaluation: a systematic literature review of the state-of-the-art mechanisms,” *Journal of Electrical Systems and Information Technology*, vol. 5, no. 3, pp. 608–622, 2018.
- [33] Y. Hu and H. Lv, “Design of Trusted BIOS in UEFI Base on USBKEY,” in *Proceedings of the 2011 International Conference On Intelligence Science And Information Engineering*, pp. 164–166, IEEE, Wuhan, China, August 2011.
- [34] D. Zhang, Z. Han, and G. Yan, “A portable TPM based on USB key,” in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, pp. 750–752, ACM, Chicago, Illinois, USA, October 2010.
- [35] B. M. Wilamowski and J. D. Irwin, *Industrial Communication Systems*, CRC Press, Boca Raton, Florida, USA, 2018.
- [36] Trusted Computing Group: TCG Specification Architecture Overview, 2007, https://trustedcomputinggroup.org/wp-content/uploads/TCG_1_4_Architecture_Overview.pdf.
- [37] T. Alves and T. Morris, “OpenPLC: an IEC 61,131-3 compliant open source industrial controller for cyber security research,” *Computers & Security*, vol. 78, pp. 364–379, 2018.
- [38] S. Fujita, K. Hata, and A. Mochizuki, “Open-PLC based control system testbed for PLC whitelisting system,” *Artificial Life and Robotics*, vol. 26, no. 4, pp. 1–6, 2020.
- [39] W. Showalter, “A universal Windows bootkit: an analysis of the MBR bootkit HDRoot,” in *Proceedings Of the 50th Hawaii International Conference On System Sciences*, pp. 6060–6068, Hawaii, USA, January 2017.
- [40] H. Gao, Q. Li, Z. Yu et al., “Research on the working mechanism of Bootkit,” in *Proceedings Of the 2012 8th International Conference on Information Science and Digital Content Technology (ICIDT2012)*, pp. 476–479, IEEE, Jeju, South Korea, June 2012.
- [41] H. J. Hu, M. Y. Fan, and G. W. Wang, “Concealment technology of Windows bootkit based on MBR,” *Journal of Computer Applications*, vol. 29pp. 83–85, Z1, 2009, in Chinese.

Research Article

A Test Cases Generation Method for Industrial Control Protocol Test

Wenli Shang ¹, Guanyu Zhang ^{2,3}, Tianyu Wang ², and Rui Zhang ³

¹School of Electronic and Communication Engineering, Guangzhou University, Guangzhou 510006, China

²Industrial Control Network and Systems Department, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China

³Information and Control Engineering Faculty, Shenyang Jianzhu University, Shenyang 110168, China

Correspondence should be addressed to Wenli Shang; shangwl@gzhu.edu.cn

Received 15 October 2020; Revised 8 January 2021; Accepted 4 March 2021; Published 13 March 2021

Academic Editor: Ting Yang

Copyright © 2021 Wenli Shang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The coverage of test cases is an important indicator for the security and robustness test of industrial control protocols. It is an important research topic to complete the test with less use cases. Taking Modbus protocol as an example, a calculation method of case similarity and population dispersion based on weight division is proposed in this paper. The method can describe the similarity of use cases and the dispersion degree of individuals in the population more accurately. Genetic algorithm is used to generate and optimize test cases, and individual similarity and population dispersion are used as fitness functions of genetic algorithm. Experimental results show that the proposed method can increase the population dispersion by 3.45% compared with the conventional methods and effectively improve the coverage of test cases.

1. Introduction

The industrial control systems control the data collection, image and sound signal processing, information transmission, and process control during the entire production process. The safety and reliability during operation are related to the stability of the entire system. In recent years, with the rapid popularization and application of computer networks, the traditional industrial control system is gradually developing towards the direction of interconnection and intelligence, and some new concepts such as Internet of things, industrial Internet of things, and industry 4.0 are proposed. However, the Internet has injected new vitality into the industrial control system but also brought the same challenges [1–3].

In security system of the industrial control system, protocol is an important guarantee for the secure transmission of information. Attacks against the protocol are one of the most common methods because of low cost, and, with the rapid development of network, remote attack becomes possible [4, 5]. As the information transmission medium of

industrial control system, it is necessary to mine possible vulnerabilities of industrial control protocol through automated testing method to ensure its security and stability.

At present, the commonly used vulnerability mining techniques are divided into static analysis, dynamic analysis, binary comparison, fuzzy testing, and so on [6–12]. Fuzzy testing has the advantages of high automation, low system consumption, low false-alarm rate, and being independent of the source code of the object program [7]. The key step in fuzzy testing is test case generation. Traditional fuzzy testing often blindly mutates a part of normal test cases when generating test cases; this blind mutation method makes the scale of test cases reach 100000 or millions, but the test effect is not ideal. Therefore, the design and improvement of test case generation strategy are one of the hot research contents of fuzzy test technology.

Test case generation algorithms for fuzzer can be divided into three categories: generation-based method, mutation-based method, and combination of the two methods [13–17]. In the current protocol testing, there are some irrationalities in the coding method and similarity determination of test

cases, which will affect the coverage of the test, and it needs to be improved. Therefore, we compared the advantages and disadvantages of the three methods, combined with the data packet structure characteristics of the test protocol, and propose a new method based on weight division to calculate the case similarity and the use case average similarity. The goal is to generate use cases with better coverage and improve test efficiency. Compared with the existing literature, this paper has the following major contributions:

- (i) A new method to determine the similarity of use cases and the concept of population dispersion are proposed, which provides a new idea and method to improve the use case coverage in the process of protocol testing.
- (ii) Different weight and distance calculation methods are set according to different protocol fields, so the similarity can be determined more accurately according to the function and data content of the use case. The change of coding method also solves the problem of inaccurate similarity judgment caused by data mutation.
- (iii) The genetic algorithm is used to generate the use case, and the similarity and the population dispersion of the case are used as the fitness function of the genetic algorithm. Automatic optimization of the use case generation is realized.

The rest of this paper is organized as follows: In Section 2, we discuss the related work. In Section 3, we provide an introduction to Modbus protocol test case design method. Section 4 is about the computing method for test cases average similarity and population dispersion. Section 5 contains simulations and results and evaluates the results based on the requirements, while Section 6 draws conclusions and reviews based on the results.

2. Related Work

2.1. Generation-Based Method. Generation-based method is to build mathematical model according to the protocol specification of test object and then generate test cases automatically. Martins et al. [18] describe a tool called ConData used as test generation for communication protocols specified as extended finite state machines. The strategy for test generation combines different specification-based test methods. Although the values for fields of interactions are automatically generated, the human intervention is always needed to determine more suitable values for test case purposes. Banks et al. [19] present SNOOZE, a tool for building flexible, security-oriented network protocol fuzzers. SNOOZE implements a stateful fuzzing approach that can be used to effectively identify security flaws in network protocol implementations. But SNOOZE is not evaluated using the code coverage metric. Li et al. [20] present an automatic vulnerability discovering method that combines automatic Protocol Reverse Engineering technology and Fuzz Testing. The method is a four-step program involving packets clustering, multiple sequences alignment, special fields recognition, and

fuzzer production, which find the structure of network packets and pursue Fuzz Testing. However, the effectiveness of the proposed method depends on the diversity of the sampling packet itself, so it is necessary to sample the network protocol multiple times and try to ensure that the network protocol is used with different parameters each time. Voyiatzis et al. [21] present the design and implementation of MTF, a Modbus/TCP Fuzzer. The MTF incorporates a reconnaissance phase in the testing procedure so as to assist mapping the capabilities of the tested device and to adjust the attack vectors towards a more guided and informed testing rather than plain random testing. The disadvantage is that Modbus/TCP Fuzzer should be redesigned for different implementations of the Modbus protocol. Liu et al. [22] proposed a heuristic network protocol fuzzy test case generation method based on the heuristic search algorithm and classification tree thought. The Peach and FTP are selected as the verification platform and target protocol, respectively. The test result verified the feasibility and effectiveness of fuzzy test case generation method of heuristic network protocol. However, the coverage of test cases in this paper depends on the accuracy of network protocol classification tree construction. Felix et al. [23] introduced a novel fuzzer, Policy Generator (PG). PG utilizes a number of heuristic techniques to improve space coverage over existing fuzzers. The empirical study demonstrates that PG generates superior coverage compared to current generation techniques. However, many of the metrics correlate and care needs to be taken when interpreting the presented data. In addition, while it is believed that the experimental framework describes this evaluation accurately, the analysis cannot be safely generalized beyond the grammatical expression of the generic firewall policy utilized in this article. Liu et al. [24] propose a vulnerability mining method combining protocol reverse analysis and fuzzy method. An improved effective counting method based on local greedy algorithm is proposed to improve the accuracy of protocol keyword extraction by 65%. Combining the lossy counting method to construct a protocol syntax tree reduces the number of spanning tree nodes by 40%. Although the performance of the proposed method is better than traditional method, it still needs to be improved in terms of operation efficiency and applicability. For example, due to the NLP method, the performance will decrease significantly while extracting keywords for pure binary protocol reverse analysis.

The main advantage of generation-based method is that the same set of test cases can be used directly for the same test objectives, and the generated test cases have high coverage [25, 26]. The main disadvantage of generation-based method is that it takes a lot of time and effort to complete the understanding of file format or protocol specification and the writing of rules. Different target types of software differ greatly. It is difficult to reuse and has a small scope of application [25, 26].

2.2. Mutation-Based Method. Mutation-based method is that a new generation of test cases is generated by mutation strategy designed based on the existing input samples. Gu et al. [27] propose a novel message matrix perturbing mode to generate test case through data mutation for application

layer protocol. Additionally, a new statistical keyword extracting technique with priority recursive splitting pattern is introduced to provide useful information for intelligent data mutation. The work presented in the paper is not perfect at several aspects. First, the static statistical analysis just finds a balance between extracting performance and computational complexity. Second, the keywords with low occurrence frequency cannot be grasped through the current method. Last but not the least, the discrimination on different protocol elements is not explicit enough for intelligent fuzzing. A test case generation technique based on mutation algorithm of precaptured IPC data is introduced in [28] in order to improve the fuzzing test efficiency. Two high-risk vulnerabilities are detected in Android 5.1.0. Analysis of these vulnerabilities highlights a critical design issue in the system services of Binder mechanism. The test case generation algorithm needs to be improved leveraging program analysis technique. Lai et al. [29] proposed a vulnerability mining method for industrial control network protocol based on fuzz testing. Protocol feature values were generated by testing cases variation factors for industrial control network protocol, each of which represented a type of ICS vulnerability features. Different test cases were generated by Modbus TCP features and variation factors. Through bypass monitoring method and Modbus TCP features relation between request and response, the difficult problem of determining the validity of testing cases was solved. However, the learning results of industrial control private protocol feature learning method will produce uncertainty due to different data sets. If the characteristics of private protocol need to be analyzed deeply, some manual analysis needs to be done. Cai et al. [30] give a fuzzy security test method based on the grammatical model and propose a grammar model for industrial control protocol based on high-order attribute grammar. The model proposes a fuzzy security test algorithm, combined with the characteristics of the industrial control protocol, and elaborates on the analysis tree structure, test case generation, and mutation strategy. The model performs comparative experiments by simulating Modbus/TCP communication which verifies that anomalous results can still be found at a lower time cost when generating fewer test cases. Accuracy of description model for the industrial control protocol based on subjective understanding will impact test case coverage. Xu et al. [31] proposed the use of deep learning technology to assist test case generation. Using the advantage of recurrent neural network to deal with character text sequences, it learnt training structure features through sample data, predicted new data that conformed to structural features, and constructed an automatic generation model to combine with random mutation algorithm. In order to make the test case generation more targeted and easier to trigger exceptions, the appropriate deep learning network should be studied to learn the auxiliary weight knowledge such as the characteristics of vulnerable points and the oriented distribution of anomalies. A fuzzing test data generation method was proposed in [32] based on dynamic construction of mutation strategy. The method was designed to use the feedback information of instrumentation to dynamically construct the

control mutation strategy and the keyword mutation strategy and to guide the fuzzer to generate test data with high coverage. However, the test effect of this method is not ideal for the target program with large input. Dynamic construction mutation method needs repeated exploration of test data and program structure. If the test data is large, it will increase the exploration time and reduce the efficiency of test data generation. Lyu et al. [33] present a novel mutation scheduling scheme MOPT, which enables mutation-based fuzzers to discover vulnerabilities more efficiently. MOPT utilizes a customized Particle Swarm Optimization (PSO) algorithm to find the optimal selection probability distribution of operators with respect to fuzzing effectiveness and provides a pacemaker fuzzing mode to accelerate the convergence speed of PSO. Yue et al. [34] present a knowledge-learn evolutionary fuzzer based on AFL, which is called LearnAFL. LearnAFL does not require any prior knowledge of the application or input format. Based on our format generation theory, LearnAFL can learn partial format knowledge of some paths by analyzing the test cases that exercise the paths. Then LearnAFL uses this format information to mutate the seeds, which is efficient to explore deeper paths and reduce the test cases exercising high-frequency paths compared to AFL.

The main advantage of the mutation-based method is that this method does not need to understand the structure and format of the current sample file, so it can be widely used [25, 26]. The main disadvantage of the mutation-based method is that it is highly dependent on the initial samples. Different initial samples will bring different code coverage, test depth, and test effect, so the efficiency is low [25, 26].

2.3. Combination of Two Methods. Hodován et al. [35] present Grammarinator, a general-purpose test generator tool that is able to utilize existing parser grammars as models. Since the model can act both as a parser and as a generator, the tool can provide the capabilities of both generation-based and mutation-based fuzzers. The presented tool is actively used to test various JavaScript engines and has found more than 100 unique issues. Grammarinator can exploit the fact that the same grammar that can generate new tests can also be used to parse existing test suites and then create new content resulting from their recombination or mutation. The tool has proven its usefulness in the hardening of real-life projects by revealing more than 100 valid unique issues. Atlidakis et al. [36] introduced Pythia, the first fuzzer that augments grammar-based fuzzing with coverage-guided feedback and a learning-based mutation strategy for stateful REST API fuzzing. Pythia's mutation strategy helps generate grammatically valid test cases and coverage-guided feedback helps prioritize the test cases that are more likely to find bugs. Pythia is the first fuzzer that augments grammar-based fuzzing with coverage-guided feedback and a learning-based mutation strategy for stateful REST API fuzzing.

A new test case generation method based on the advantages of the above methods is proposed in this paper. Firstly, the characteristics of general transmission message of

industrial control protocol are analyzed, test cases are designed based on the construction of description model, and coding method of use cases is designed for genetic algorithm. Secondly, genetic algorithm is used to generate and optimize use cases, which realizes the automatic iteration and update of use case population. Finally, in order to improve test coverage and vulnerability discovery rate, the concept of dangerous point is proposed, and, based on this, a composite fitness function is designed to monitor and adjust the state of use case population.

3. Modbus Protocol Test Cases Design

3.1. Message Feature Analysis and Encoding. Choosing appropriate encoding method of use cases for protocol testing can reduce the time complexity of generating test cases and complete the conversion from encoding files to data packets faster. Figure 1 shows the data fields contained in the data packets of Modbus communication protocol and the byte length of each field [37].

In Modbus protocol packets, because the transmission identifier and protocol identifier are independent of the packets' content, these two fields cannot be considered when constructing test cases [38], so each test case can be mathematically expressed as in the following equation:

$$\text{case} = [l \ u \ f \ d], \quad (1)$$

where l is the length of the data field, and its value matches the data length contained in the following three fields. u is the address identifier, and value range is 0 to 255. f is the function code, which is divided into public function code and user-defined function code in Modbus, and its value range is 1 to 127. d is a data field, and the data information of this field depends on the function code.

When encoding test cases, binary encoding is the most common encoding method, and Hamming distance can be used to measure similarity between two test cases, as shown in the following equation:

$$d(A, B) = \sum_{i=0}^n A_i \oplus B_i, \quad (2)$$

where A_i and B_i denote the i -th characters of the strings A and B ; \oplus means to judge whether A_i and B_i are the same; when they are the same, $A_i \oplus B_i = 0$; when they are not, $A_i \oplus B_i = 1$.

However, when comparing the similarity of two test cases to calculate the Hamming distance, the Hamming cliff problem may occur [39]. Therefore, Gray code is used in this paper, which can effectively avoid the Hamming cliff problem and realize a more accurate description of the similarity of protocol packets. Assuming that there is a binary code of $B = B_n B_{n-1} B_{n-2} \dots B_1 B_0$ and its corresponding Gray code is $G = G_n G_{n-1} G_{n-2} \dots G_1 G_0$, then the value of the two codes satisfied the following equation:

| Transaction identifier | Protocol identifier | Length | Unit identifier | Function code | Data |
|------------------------|---------------------|----------|-----------------|---------------|-------|
| Byte 0/1 | Byte 2/3 | Byte 4/5 | Byte 6 | Byte 7 | Other |

FIGURE 1: Modbus data packets structure.

$$\begin{cases} G_i = B_i, & i = n, \\ G_{i-1} = B_{i-1} \oplus B_i, & i = 1, 2, \dots, n-1, \end{cases} \quad (3)$$

where G_i are the i -th bits of binary code and Gray code and \oplus is XOR operation.

Figure 2 shows the effect of clustering on the same set of data when calculating distance using two different encoding methods. It can be seen from the figure that some data may not be able to find the cluster center accurately when using binary code (Figure 2, left) to calculate the distance, while Gray code (Figure 2, right) can effectively avoid this problem.

In summary, Gray code avoids the Hamming cliff problem in binary coding, so the similarity between two Gray-coded strings can be described by the number of different bits, namely, Hamming distance.

3.2. Method for Calculating Similarity of Test Cases with Weights. In the Modbus protocol, the length of each field of the message sequence is basically fixed, but the length of the data storage field is dynamic, and the function of each field and the impact on the security of the message are different. Some fields are related to each other. If the Hamming distance is directly used as the similarity determination between the encoded strings of the two test cases, there is a certain irrationality.

In order to solve these problems, a weight distance calculation method based on internal classification is proposed in this paper. The weight of different fields is set in different value, and the distances of different fields are calculated according to corresponding functions. The data segment is special, because it is related to other fields, and a unique design is required to calculate the relevant distance. The weight coefficient of each field is determined by Analytic Hierarchy Process (AHP).

Assuming that there are test cases A and B , first calculate the corresponding distances of each functional field of them, then combine the weights of the fields, and calculate their overall similarity. The final calculation formula is shown in following equation:

$$\text{dis}_{AB} = \sum_{i=0}^4 w_i \cdot d(A_{vi}, B_{vi}), \quad (4)$$

where $w = [w_1, w_2, w_3, w_4]$ is the weight of each field. A_{vi} and B_{vi} are the corresponding fields of the two test cases, and $d(A_{vi}, B_{vi})$ is the distance between the two corresponding fields. The distance calculation method for different fields is slightly different.

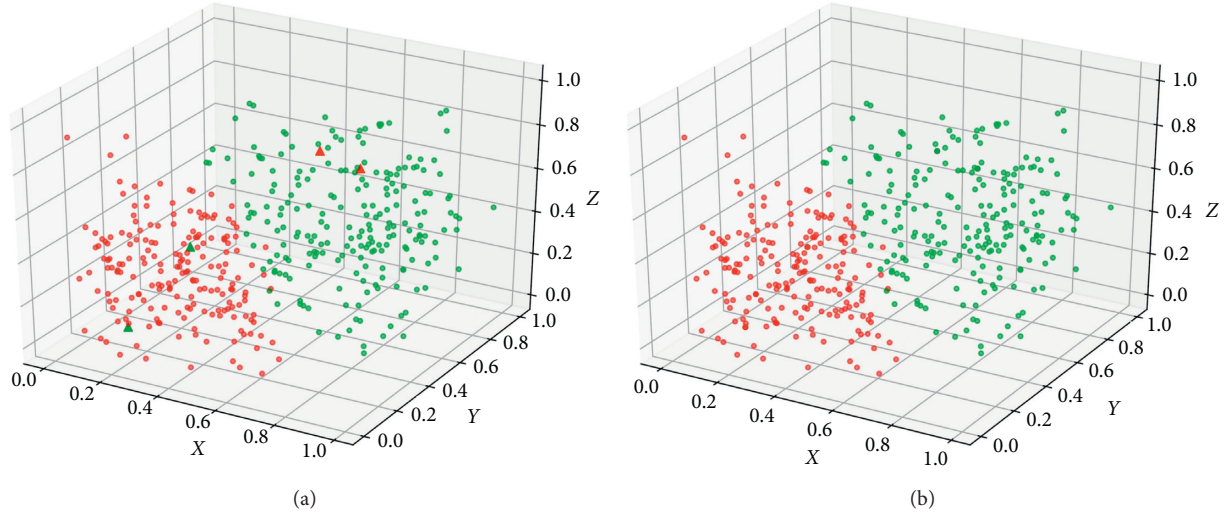


FIGURE 2: Clustering results of two encoding methods.

The pairwise comparison matrix determined by Analytic Hierarchy Process is shown in the following equation:

$$A = \begin{bmatrix} 1 & 3 & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{3} & 1 & \frac{1}{5} & \frac{1}{3} \\ 3 & 5 & 1 & 3 \\ 2 & 3 & \frac{1}{3} & 1 \end{bmatrix}. \quad (5)$$

Consistency of the pairwise comparison matrix was checked. If test coefficient $CR = 0.0386 < 0.9$, then consistency check is passed. The calculated weight of each field is shown in the following equation:

$$w = [0.1682 \quad 0.0769 \quad 0.5167 \quad 0.2382]. \quad (6)$$

According to the characteristics of the Modbus test cases, the length of length field, address identifier field, and function code field are fixed, while the length of the data field is dynamically variable and is associated with other fields. Therefore, when calculating the distance between the corresponding fields of the two use cases, two different methods are used to calculate the distance of the fixed-length and variable-length fields. For fixed-length fields, the Hamming distance can be directly used.

The length of the data field is dynamically variable. When describing the distance, Hamming distance will have a large deviation, and Levenshtein distance can solve this problem. Levenshtein distance is to find the minimum number of transformations required to convert string A to string B. It can more describe the difference between two strings of different lengths accurately. The calculation method is shown in the following equation:

$$\text{lev}_{A,B}(i, j) = \begin{cases} \max(i, j), & \min(i, j) = 0, \\ \min \begin{cases} \text{lev}_{A,B}(i-1, j) \\ \text{lev}_{A,B}(i, j-1) \\ \text{lev}_{A,B}(i-1, j-1) + 1_{A_i \neq B_j} \end{cases}, & \min(i, j) \neq 0, \end{cases} \quad (7)$$

where i and j are the subscripts of string A to string B. $\max(i, j)$ is the maximum value. $\min(i, j)$ is the minimum value.

Therefore, the similarity calculation equation (4) of the two test cases can be further optimized into the following equation:

$$\text{dis}_{AB} = \sum_{i=1}^3 w_i \cdot d(A_{vi}, B_{vi}) + \text{lev}_{A_d, B_d}(m, n), \quad (8)$$

where $\text{lev}_{A_d, B_d}(m, n)$ is the Levenshtein distance between the two data fields of m and n .

4. Average Similarity and Population Dispersion of Test Cases

In the test case generation process, the iteration is based on the population, so it is necessary to describe first-generation population from the perspective of the whole population. Here, the average similarity of population test cases is designed to describe the population state. The average similarity of test cases refers to the overall degree of dispersion among individuals in a population. When the average similarity of test cases is low, it means that the overall similarity of individuals within the population is too high, and the coverage of test cases is low [40]. At this time, the parameter information in the test cases generation process, such as the mutation probability and the similarity threshold, can be appropriately changed to adjust the

distribution of the generated test cases and improve the coverage of the test cases.

When describing the average similarity of test cases of individuals in the entire population, it can be described by the average distance between individuals. This method is feasible to some extent, but each individual needs to calculate the distance between itself and all other individuals. As a result, this method has a lot of repeated calculation and low efficiency. In addition, if an extremely uniform edge distribution occurs, it will also lead to misjudgment. Therefore, the concept of average similarity of test cases is proposed in this paper, and a new calculation method is designed to accurately reflect the distribution of individuals in the population and reduce the amount of calculation.

Firstly, values of individual fields in the population are normalized, which is expressed mathematically in the following equation:

$$v_n = \frac{c_n - c_{n_min}}{c_{n_max} - c_{n_min}}, \quad (9)$$

where c_{n_max} is the maximum value of the field in the population; c_{n_min} is the minimum value of the field in the population.

The sum of each field is averaged to calculate the mean center test case, as shown in equation (10), and the calculation method of each field is as in equation (11).

$$\overline{case} = [\overline{l_v} \ \overline{u_v} \ \overline{f_v} \ \overline{d_v}], \quad (10)$$

$$\overline{v} = \frac{1}{m} \cdot \sum_{i=1}^m v_i, \quad (11)$$

where m is the total number of test cases in the population and v_i is the current field of the test cases.

The similarity between the test cases and the central test case can be used to indicate the outlier degree of the test cases, as shown in the following equation:

$$s = \sum_{i=1}^3 w_i \cdot d(A_{vi}, \overline{C_{vi}}) + lev_{A_{vi}, \overline{C_{vi}}}(m, n). \quad (12)$$

The calculation time complexity of the average similarity of the test cases is $2n$; compared with the time complexity $n \lg n$ of the general method, there will be a significant efficiency improvement when n is larger. Then the dispersion of the whole population can be described by the following equation:

$$sca = \frac{1}{n} \cdot \sum_{i=1}^n s_i. \quad (13)$$

5. Experimental Evaluation

By designing the encoding method and the similarity calculation method between test cases, combined with the description of the average similarity of test cases in the test cases population, theoretically, it can effectively improve the efficiency of test cases generation and increase the coverage of test cases. In order to verify the correctness of the proposed method, a set of comparative experiments are

designed, and genetic algorithm is used as the core algorithm for test case generation. The encoding method, individual similarity, and average similarity of test cases are calculated by the proposed method and the conventional method, respectively, and the test cases generated by the two methods are compared and analyzed.

Genetic algorithm is an intelligent optimization algorithm, which is often used to find the global optimal solution, and we adjust the population optimization direction by designing the corresponding fitness function. In the test case generation method designed in this paper, the population convergence direction of genetic algorithm is a suspicious case in historical data. Suspicious test cases are cases that cause test target anomalies during the test process. Taking these cases as the convergence center of next genetic algorithm can effectively reduce the randomness of test case generation. These test cases are called ‘‘suspicious points.’’ Based on this, the fitness function of the genetic algorithm designed for two sets of experiments is shown in the following equation:

$$\begin{cases} f_p(s_A) = 1 - \frac{s_A}{s_{max}}, & \text{suspicious points exist,} \\ f_p(A) = 1 - \frac{\text{dis}(dp_p, A_i)}{\max(\text{dis}(dp_p, A_i))}, & \text{else,} \end{cases} \quad (14)$$

where dis is the similarity between the test case and the suspicious point; the calculation method is shown in formula (3). s_A is the average similarity of the test case.

The meaning of fitness function is that when there are suspicious points in the population, the population converges to the suspicious case. When there is no suspicious point, the population with higher average similarity of test cases is preferred. Other parameter settings of genetic algorithm are mutation probability $P_m = 0.2$ and crossover probability $P_c = 0.6$.

The whole experimental procedure designed is shown in Figure 3. Firstly, the initial test case population for the two experiments is constructed manually, and the initial population is encoded according to the encoding method mentioned above. Secondly, the initial population is input into the test case generation module, and two different fitness function calculation methods are used to generate and optimize the test cases. Finally, the result monitoring module records the operation results.

The script development language of the experiment is *Python 3*, and Modbus communication simulation software used in the test is Modbus Poll and Modbus Slave. Firstly, Modbus Poll is used to establish data communication with Modbus Slave, Wireshark packet capture tool is used to obtain normal communication messages, and representative data messages are selected to analyze the data characteristics and construct the initial population. Secondly, the initial population is sent to the test cases generation and optimization module to iterate, optimize, and update test cases. Finally, each generation of population is sent to the target for testing. Statistical analysis was performed on the test cases

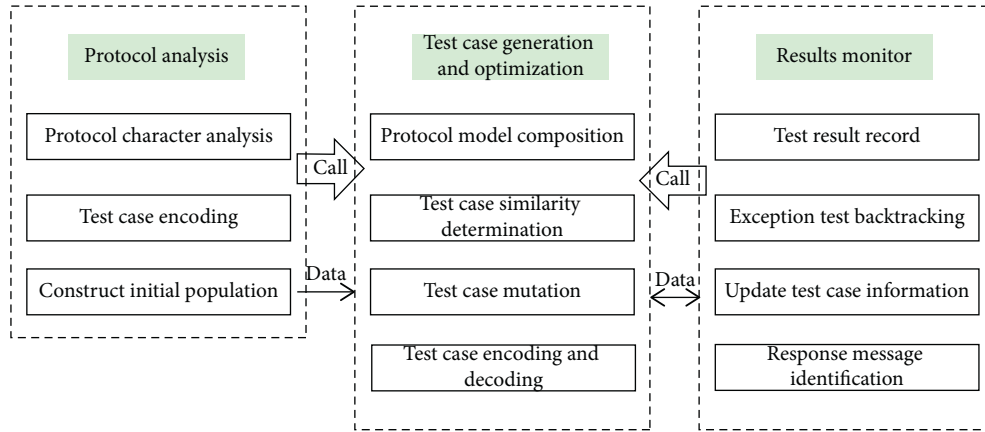


FIGURE 3: The whole experimental procedure designed.

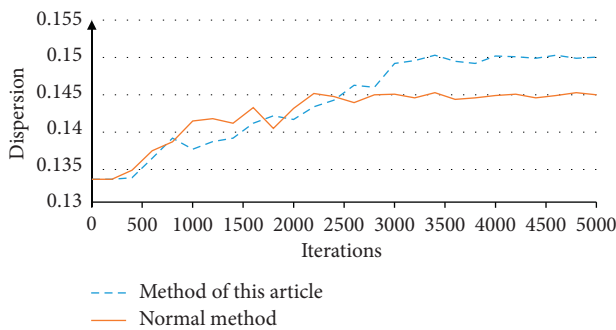


FIGURE 4: Trend of population dispersion.

data generated by the two methods. During the experiment, the average similarity of the first 5000 generations of population test cases was calculated. The results are shown in Figure 4.

In two groups of experiments using different methods, during the population iteration process, the dispersion gradually increased and eventually stabilized. At the beginning of the experiment, since the same initial population was used, the dispersions of two groups were the same. However, with the iteration of the population, when both of them are stable, the dispersion of the population produced by the improved method is 3.45%, which is higher than that of the conventional method. It is generally believed that the higher the dispersion between individuals within a population, the higher the coverage of test cases [21]. Therefore, it can be considered that the coverage of test cases generated by the improved method is higher than the conventional method, and it also proves that the method proposed in this article has certain advantages over the conventional method. Based on the proposed method, we design a fuzzy tester [41].

6. Conclusion

A new test cases similarity determination method and the concept of population dispersion are proposed in this paper, which provides a new idea and method for improving the test cases coverage in the protocol testing process. In the determination of test cases similarity, different weights and distance calculation methods are set according to different

protocol fields, which can more accurately determine the similarity according to the function of the test cases and data content, and the change of the encoding method effectively resolves the problem of inaccurate similarity determination caused by data mutation. The genetic algorithm is introduced into the test cases generation algorithm, and the test cases similarity and population dispersion are used as the basis for constructing the fitness function of the genetic algorithm, and the automatic optimization of the test cases generation is realized. The test cases data generated in the experiment shows the effectiveness of the method. Our planned future work is twofold. First, we plan to improve the applicability of the method and apply it to the generation of test cases for other protocols. Second, we plan to optimize the time complexity of the algorithm.

Data Availability

The data used to support the findings of this study have not been made available because the generated test cases were not backed up in time.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by “National Key R&D Program of China” (2018YFB2004200), the open project of Zhejiang Lab “Construction Technology of Local High Security Trusted Execution Environment for Edge Intelligent Controller” (2021KF0AB06), and the National Natural Science Foundation of China “Research on anomaly detection and security awareness method for industrial communication behaviours” (61773368).

References

- [1] O. E. Idrissi, A. Mezrioui, and A. Belmekki, “Cyber security challenges and issues of industrial control systems—some security recommendations,” in *Proceedings of the IEEE*

- International Smart Cities Conference (ISC2)*, pp. 330–335, Casablanca, Morocco, April 2019.
- [2] M. R. Asghar, Q. Hu, and S. Zeadally, “Cybersecurity in Industrial control systems: issues, technologies, and challenges,” *Computer Networks*, vol. 165, Article ID 106946, 2019.
 - [3] T. Miyachi and T. Yamada, “Current issues and challenges on cyber security for industrial automation and control systems,” in *Proceedings of the SICE Annual Conference (SICE)*, pp. 821–882, Sapporo, Japan, October 2014.
 - [4] D. Myers, K. Radke, S. Suriadi, and E. Foo, “Process discovery for industrial control system cyber attack detection,” *ICT Systems Security and Privacy Protection 2017, Proceedings (IFIP Advances in Information and Communication Technology)*, Springer, vol. 502, pp. 61–75, Switzerland, 2017.
 - [5] C. Lin, S. Wu, and M. Lee, “Cyber attack and defense on industry control systems,” in *Proceedings of the IEEE Conference on Dependable and Secure Computing*, pp. 524–526, Taipei, Taiwan, August 2017.
 - [6] S. M. Ghaffarian and H. R. Shahriari, “Software vulnerability analysis and discovery using Machine-Learning and Data-Mining techniques,” *ACM Computing Surveys*, vol. 50, no. 4, pp. 1–36, 2017.
 - [7] Y. X. Lai, H. Gao, and J. Liu, “Vulnerability mining method for the modbus TCP using an anti-sample fuzzer,” *Sensors*, vol. 20, no. 7, p. 2040, 2020.
 - [8] C. Wang, Q. Li, X. H. Wang et al., “An android application vulnerability mining method based on static and dynamic analysis,” in *Proceedings of the IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, IEEE, June 2020.
 - [9] T. Tu, H. Zhang, B. Qin et al., “A vulnerability mining system based on fuzzing for IEC 61850 protocol,” *Advances in Engineering Research (AER) in Proceedings of the 5th international conference on frontiers of manufacturing science and measuring technology (FMSMT 2017)*, vol. 130, pp. 589–597, Taiyuan, China, June 2017.
 - [10] W.-N. Kim, M.-S. Jang, J. Seo, and S. Kim, “Vulnerability discovery method based on control protocol fuzzing for a railway SCADA system,” *The Journal of Korea Information and Communications Society*, vol. 39C, no. 4, pp. 362–369, 2014.
 - [11] S. J. Kim and T. Shon, “Field classification-based novel fuzzing case generation for ICS protocols,” *Journal of Supercomputing*, vol. 74, no. 9, 2018.
 - [12] T. Wang, Q. Xiong, H. Gao et al., “Design and implementation of fuzzing technology for OPC protocol,” in *Proceedings of the Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 424–428, Beijing, China, October 2013.
 - [13] X. Zhang and Z. J. Li, “Overview of fuzzy testing technology,” *Computer Science*, vol. 43, no. 5, pp. 1–8, 2016, in Chinese.
 - [14] T. L. Munea, H. Lim, and T. Shon, “Network protocol fuzz testing for information systems and applications: a survey and taxonomy,” *Multimedia Tools and Applications*, vol. 75, no. 22, pp. 14745–14757, 2016.
 - [15] T. Kitagawa, M. Hanaoka, and K. Kono, “AspFuzz: a state-aware protocol fuzzer based on application-layer protocols,” in *Proceedings of the IEEE Symposium on Computers and Communications*, pp. 202–208, Riccione, Italy, June 2010.
 - [16] Y. J. Zhang, Z. J. Li, X. K. Liao et al., “Survey of automated whitebox fuzz testing,” *Computer Science*, vol. 41, no. 2, pp. 7–10, 2014.
 - [17] M. B. Cohen, J. Snyder, and G. Rothermel, “Testing across configurations,” *ACM SIGSOFT Software Engineering Notes*, vol. 31, no. 6, pp. 1–9, 2006.
 - [18] E. Martins, S. B. Sabiao, and A. M. Ambrosio, “ConData: a tool for automating specification-based test case generation for communication systems,” in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, vol. 8, Maui, HI, USA, January 2000.
 - [19] G. Banks, M. Cova, V. Felmetzger et al., “SNOOZE: Toward a stateful network protocol fuzzer,” in information security,” in *Proceedings of the International Conference, Isc, Samos Island, Greece, Samos Island, Greece, August 2006*.
 - [20] W.-M. Li, A.-F. Zhang, J.-C. Liu, and Z.-T. Li, “An automatic network protocol fuzz testing and vulnerability discovering method,” *Chinese Journal of Computers*, vol. 34, no. 2, pp. 242–255, 2011, in Chinese.
 - [21] A. G. Voyiatzis, K. Katsigiannis, and S. Koubias, “A Modbus/TCP Fuzzer for testing internetworked industrial systems,” in *Proceedings of the IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, pp. 1–6, Berlin, Germany, September 2015.
 - [22] J. J. Liu and Y. D. Yuan, “Research on network protocol fuzzy test case generation method based on heuristic search and classification tree,” *Modern Electronics Technique*, vol. 39, no. 21, pp. 36–39, 2016, in Chinese.
 - [23] A. Felix, A. F. Tappenden, and J. Miller, “Policy generator (PG): a heuristic-based fuzzer,” in *Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS)*, pp. 5535–5544, Koloa, HI, USA, March 2016.
 - [24] H. X. Wang, C. Y. Zhu, H. Ying et al., “A fuzzy testing method of industrial control protocol based on reverse analysis,” *Electric Power Information and Communication Technology*, vol. 17, no. 4, pp. 5–13, 2019, in Chinese.
 - [25] C. Miller and Z. Peterson, “Analysis of mutation and generation-based fuzzing,” 2007, <https://www.defcon.org/images/defcon-15/dc15-presentations/Miller/Whitepaper/dc-15-miller-WP.pdf>.
 - [26] K. Chen, C. Song, L. M. Wang et al., “Using memory propagation tree to improve performance of protocol fuzzer when testing ICS,” *Computers & Security*, vol. 87, Article ID 101582, 2019.
 - [27] S. J. Gu, Y. Y. Song, X. Zhao et al., “Fuzzing test data generation based on message matrix perturbation with keyword reference,” in *Proceedings of the IEEE MILCOM 2011 Military Communications Conference*, pp. 1115–1120, Baltimore, MD, USA, November 2011.
 - [28] K. Wang, Y. Q. Zhang, Q. X. Liu et al., “A fuzzing test for dynamic vulnerability detection on Android Binder mechanism,” in *Proceedings of the IEEE Conference on Communications and Network Security (CNS)*, pp. 709–710, Florence, Italy, December 2015.
 - [29] Y. X. Lai, K. X. Yang, J. Liu et al., “Vulnerability mining method for industry control network protocol based on fuzzing test,” *Computer Integration Manufacturing System*, vol. 25, no. 9, pp. 2265–2279, 2019, in Chinese.
 - [30] J. Cai, Q. Li, Y. Chen, Y. Liu, Y. Xia, and S. Rahmany, “Troubleshooting test method based on industrial control grammar model,” in *Proceedings of the IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pp. 404–409, New York, NY, USA, August 2019.

- [31] P. Xu, J. Y. Liu, B. Lin et al., “Generation of fuzzing test case based on recurrent neural networks,” *Application Research of Computers*, vol. 36, no. 9, pp. 2679–2685, 2019, in Chinese.
- [32] L. L. Jiao, S. L. Luo, W. Cao et al., “Fuzzing test data generation method based on dynamic construction of mutation strategy,” *Transactions of Beijing Institute of Technology*, vol. 39, no. 5, pp. 539–544, 2019, in Chinese.
- [33] C. Y. Lyu, S. L. Ji, C. Zhang et al., “MOPT: optimized mutation scheduling for fuzzers,” in SEC’19,” in *Proceedings of the 28th USENIX Conference on Security Symposium*, pp. 1949–1966, Berkeley, CA; USA, August 2019.
- [34] T. Yue, Y. Tang, B. Yu, P. Wang, and E. Wang, “Learn AFL: greybox fuzzing with knowledge enhancement,” *Institute of Electrical and Electronics Engineers Access*, vol. 7, pp. 117029–117043, 2019.
- [35] R. Hodován, K. Kiss, and T. Gyimóthy, “Grammarinator: a grammar-based open source fuzzer,” in *Proceedings of the 9th ACM SIGSOFT International Workshop on Automating TEST Case Design, Selection, and Evaluation*, pp. 45–48, Lake Buena Vista, FL, USA, November 2018.
- [36] V. Atlidakis, R. Geambasu, P. Godefroid et al., “Pythia: Grammar-Based Fuzzing of REST APIs with Coverage-Guided Feedback and Learning-Based Mutations,” 2020, <https://arxiv.org/pdf/2005.11498v1.pdf>.
- [37] I. N. Fovino, A. Carcano, M. Masera et al., “Design and implementation of a secure modbus protocol,” in *Proceedings of the International Conference on Critical Infrastructure Protection*, pp. 33–36, Springer, Arlington, VA, USA, March 2009.
- [38] J. Luswata, P. Zavorsky, B. Swar et al., “Analysis of SCADA security using penetration testing: a case study on Modbus TCP protocol,” in *Proceedings of the 29th Biennial Symposium on Communications (BSC)*, Toronto, CA, USA, June 2018.
- [39] R. A. Caruana and J. D. Schaffer, “Representation and hidden bias: Gray vs. Binary coding for genetic algorithms,” in *Proceedings of the Fifth International Conference on Machine Learning*, pp. 153–161, Ann Arbor, MI, USA, June 1988.
- [40] A. Arrieta, S. Wang, U. Markiegi et al., “Search-based test case generation for Cyber-Physical Systems,” in *Proceedings of the Institute of Electrical and Electronics Engineers Congress on Evolutionary Computation (CEC)*, pp. 688–697, Donostia-San Sebastian, Spain, June 2017.
- [41] G. Zhang, W. Shang, B. Zhang, C. Chunyu, and Z. Rui, “Fuzzy test method for industrial control protocol combining genetic algorithm,” *Computer Application Research*, vol. 38, no. 3, 2021, in Chinese.

Research Article

An Algorithm of Occlusion Detection for the Surveillance Camera

Peng Shi , Bin Hou , Jing Chen , and Yunxiao Zu 

School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Yunxiao Zu; zuyx@bupt.edu.cn

Received 11 October 2020; Revised 30 January 2021; Accepted 15 February 2021; Published 22 February 2021

Academic Editor: Ting Yang

Copyright © 2021 Peng Shi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As more and more surveillance cameras are deployed in the Internet of Things, it takes more and more work to ensure the cameras are not occluded. An algorithm of detecting whether the surveillance camera is occluded is proposed by comparing the similarity of the images in this paper. Firstly, the background modeling method based on frame difference is improved. The combination method of the background difference and frame difference is proposed, and the experimental results showed that the combination algorithm can extract the background image of the video more quickly and accurately. Secondly, the LBP (Local Binary Patterns) algorithm is used to compare the similarity between the background image and the reference image. By changing the window size of the LBP algorithm and setting an appropriate threshold, the actual demands can be satisfied. So, the algorithms proposed in this paper have high application value and practical significance.

1. Introduction

In the context of the Internet of Things and communication technology being ever-changing, from smart home to smart city, the coverage of the Internet of Things is getting wider and wider. There are more and more surveillance cameras deployed in the Internet of Things. These surveillance cameras are closely related to many fields of our life and work. The surveillance cameras have many functions, such as live watching, video watching, and abnormal warning, which are very important for maintaining personal and social security. However, the camera will be occluded due to various accidents or human factors. For example, some criminals and suspicious people deliberately occlude the cameras in order to avoid being caught [1, 2]. Therefore, it has very important application value and practical significance to ensure the surveillance camera is not occluded.

At present, the methods detecting whether the surveillance camera is occluded are mainly based on the difference between frames [3, 4]. This kind of method is aimed at the monitoring image changing significantly in a short time when the camera is occluded, so it can detect whether the camera is occluded by comparing the difference between frames. But any error will lead to an inaccurate detection

result; the rates of false negatives and false positives are high and the application range is small.

In actual application, the scene monitored by the same camera is unchanged, so the monitoring image can be divided into two parts: the unchanged background image and the changed foreground image. Based on this feature, the methods [5, 6] that determine whether the camera is occluded or not are proposed by many scholars. These methods mainly measure the difference between the current frame image and the reference image by using some appropriate image feature vectors. However, the presence of foreground in the current frame image would have a certain impact on the detection results.

In view of the above methods' shortcoming, the existing background modeling method based on frame difference is improved in this paper by combining the background difference. This improved method can be used to extract the background image of the video more quickly and accurately. Compared with other image features, LBP (Local Binary Patterns) feature is easier to extract, the calculation of the LBP feature is simpler, and the accuracy of the LBP feature is higher. Therefore, the LBP feature is used to construct a feature vector to measure the similarity between the background image and the reference image in this paper. And

whether the surveillance camera is occluded can be determined according to the similarity. The calculation of occlusion detection based on comparison of image similarity is simple, so it is easy to be implemented. Occlusion detection based on comparison of image similarity not only effectively eliminates the influence of foreground but also is robust to illumination.

2. Materials and Methods

2.1. The Principle of the Background Modeling Algorithm Based on Background Difference and Frame Difference. In machine vision, background modeling is the basic technology of video processing. In recent years, a large number of background modeling methods have been proposed by many experts and scholars. Currently, the widely used background modeling methods mainly include median background modeling method [7], mean background modeling method, Gaussian distribution background modeling method [8], and ViBe algorithm [9]. Not only do these algorithms require a large number of video frames for background modeling but also the calculation is complex and requires a long time to extract the video background, which is difficult to meet the real-time requirements when detecting whether the surveillance camera is occluded.

The frame difference method is the most primitive and the simplest background modeling method. The frame difference method can be implemented quickly and it has a wide range of applications. In literature [10], an improved frame difference method is proposed for background modeling of video. Combined with the idea of time series statistics, the background model is established by counting the number of continuous frames. If the gray value of a pixel changes little for several continuous frames, the gray value is considered as the gray value of the background at the point. If the gray value of a pixel changes a lot between two adjacent frames, the gray value is considered as the gray value of the foreground. The gray value of the two adjacent frames will continue to be compared until the gray value of the background at this point can be determined. For the background point whose gray value cannot be determined for a long time, the gray value of the pixel point in the last frame is taken as the gray value of the background point. Because the area is small, it has little influence on the background model.

The above-improved frame difference method still needs a certain number of video frames for background modeling. So the real-time requirements still cannot be satisfied when detecting whether the surveillance camera is occluded. In this paper, the background modeling method is further improved by combining the background difference method. The gray value of the current frame is compared with the gray value of the reference image at the same pixel point. If the difference is small, the gray value of the current frame is considered as the gray value of the background at this point. If the difference is large, the improved frame difference

method in literature [10] is used to determine the gray value of the background at this point.

In the improved frame difference method in literature [10], determining whether the gray value of every pixel point is the gray value of the background at this point needs at least *Thread2* frames (*Thread2* is an adaptive threshold). In the background modeling algorithm based on background difference and frame difference, determining whether the gray value of every pixel point is the gray value of the background at this point only needs at least 1 frame. So, the background modeling algorithm based on background difference and frame difference is easier to meet the real-time requirements.

2.2. The Process of the Background Modeling Algorithm Based on Background Difference and Frame Difference. The specific process of the background modeling algorithm based on background difference and frame difference is shown in Figure 1.

Img1 is the previous frame image, *Img2* is the current frame image, *BImg* is the background image which needs to be built, *RImg* is the reference image extracted from an unoccluded surveillance video, *BimgFlag* is the number of continuous frames whose gray value change little at the same pixel point, *Flag* is the number of pixels whose gray value is not determined in the background image. In the background modeling algorithm based on background difference and frame difference, there are three adaptive thresholds *Thread1*, *Thread2*, and *Thread3*: *Thread1* is the difference threshold of two frames, *Thread2* is the number threshold of continuous frames whose gray value changes little at the same pixel point, and *Thread3* is the number threshold of pixels whose gray value is not determined in the background image. The specific process is as follows:

- (1) Initialize: The first frame image *Img1* and the reference image *RImg* are loaded, and both the values of *BImg* and *BimgFlag* are set as the full-zero matrix. The value of *Flag* is set as the number of pixels in the image *Img1*.
- (2) Judge the number threshold of continuous frames: a new frame image *Img2* is loaded. For every pixel point in *Img2*, if $BimgFlag(x) < Thread2$, let $DiffImg(x) = |Img2(x) - RImg(x)|$ and turn to Step 3. Otherwise, Step 2 will be performed to continue the iteration.
- (3) Judge the difference threshold between the current frame and the reference image: for each pixel that meets the condition in Step 2, if $DiffImg(x) < Thread1$, let $BimgFlag(x) = Thread2$, $BImg(x) = Img2(x)$, $Flag = Flag - 1$, and turn to Step 5. Otherwise, let $DiffImg(x) = |Img2(x) - Img1(x)|$ and turn to Step 4.
- (4) Judge the difference threshold between the current frame and the previous frame: for each pixel that

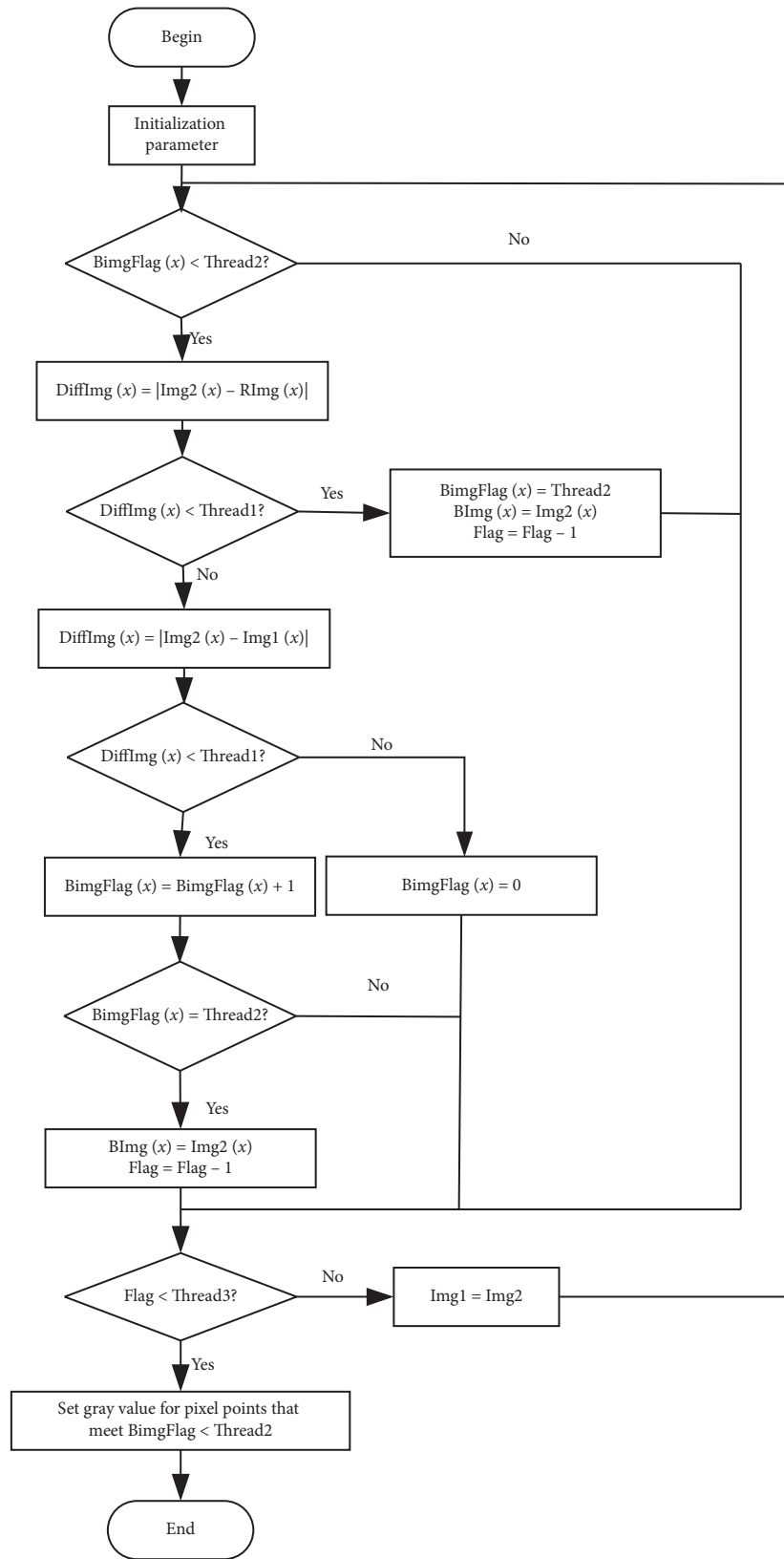


FIGURE 1: The specific process of the background modeling algorithm.

meets the condition in Step 3, if $\text{DiffImg}(x) < \text{Thread1}$, let $\text{BimgFlag}(x) = \text{BimgFlag}(x) + 1$. Otherwise, let $\text{BimgFlag}(x) = 0$. If $\text{BimgFlag}(x) = \text{Thread2}$, let $\text{BImg}(x) = \text{Img2}(x)$, $\text{Flag} = \text{Flag} - 1$.

- (5) Judge the iterative condition: When $\text{Flag} < \text{Thread3}$, it shows that the constructed background image has met the requirements. Then, the gray value of the pixel points which meet $\text{BimgFlag} < \text{Thread2}$ in the background image will be set. Otherwise, let $\text{Img1} = \text{Img2}$ and turn to Step 2, and the iteration is done.

In the background modeling algorithm based on background difference and frame difference, the threshold Thread1 is the key factor to determine the gray value of the background pixel point. Its value depends on the difference in the gray value between the background and the foreground. The threshold Thread2 mainly depends on the speed of the object moving in the video. The slower the object moves, the larger the value should be. The threshold Thread3 is the condition of iteration, its value is related to the micromovements in the video. The value of Thread3 affects the number of iterations and the quality of training. By setting the thresholds, the noise interference can be removed to a large extent and the background image can be obtained more ideal.

2.3. The Principle of LBP Algorithm. There are many features that can be used to measure the differences between different images, including gray histogram, edge histogram, color histogram, corner feature, and scale invariant feature [11]. Through the analysis and study, it is found that there is a certain difference in texture features of the background between the occluded video and the unoccluded video. Therefore, the LBP algorithm (local binary mode) is selected to measure the difference of different background images in this paper. The LBP algorithm is widely used in face recognition [12], facial expression recognition [13], image retrieval [14], image classification [15], and other fields, and it has achieved good results. Compared with other simple methods of feature extraction, the recognition accuracy of the LBP algorithm is higher. Compared with other methods with high recognition accuracy, the calculation of the LBP algorithm is more simple and the LBP feature is easier to extract. Based on the above characteristics, the requirements of real-time and accuracy can be better met by using the LBP algorithm when detecting whether the surveillance camera is occluded.

The LBP algorithm is not only a nonparameter algorithm to describe the difference of the gray value between the center pixel and its neighborhood pixels in the image, but also an efficient algorithm that describes local texture features. The original LBP operator takes the gray value of the central pixel point as the threshold in the window of 3×3 . The gray values of eight neighborhood pixel points are

compared with the threshold. If the gray value of the neighborhood pixel point is greater than the threshold, the coding value of the neighborhood point is 1. Otherwise, the coding value of the neighborhood point is 0. Then, the coding value of each neighborhood pixel point is assigned weight 2^i , $i = 0, 2, \dots, 7$. Through the above coding, the coding values of the eight neighborhood pixel points can form into an eight-bit binary. The decimal value represented by the binary is the LBP value that we want to find. The LBP value can effectively reflect the texture information in the window and it will be used to replace the gray value of the original center pixel points.

As shown in Figure 2, in the window of 3×3 , it is assumed that the gray value of the center pixel is p_c and the gray values of eight neighborhood pixels are $p_i, i \in [0, 7]$. If $p_i > p_c$, the coding value of p_i is 1. Otherwise, the coding value of p_i is 0. The calculation process of LBP value is as follows:

$$\text{LBP}_{P,R} = \sum_{i=0}^{P-1} s(p_i - p_c)2^i. \quad (1)$$

In formula (1), P is the number of neighborhood pixel points, its value is 8. s is the symbolic function:

$$s(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases} \quad (2)$$

The LBP algorithm only subtracts the gray values of the central pixel points and the neighborhood pixel points in the selected window. It can simply and quickly extract the local texture feature of the image without a complex learning process. So, the calculation of the LBP algorithm is simple and the range of LBP algorithm's applications is wide.

2.4. The Algorithm of Occlusion Detection for the Surveillance Camera. First, the improved frame difference method in literature [10] is used to extract the reference image by using an unoccluded surveillance video. Then, the background modeling method based on background difference and frame difference is used to extract the background image of the video which needs to be detected. The LBP algorithm is used to dispose the reference image and the background image, respectively, and the image disposed by the LBP algorithm is called mapping. In practical application, the blocked histogram of the mapping is used to construct the feature vectors. The feature vectors are compared by using the nonparametric method to measure the difference between images. There are many nonparametric methods that can be used to compare the difference of two histograms, such as Euclidean distance, chi-square statistics, Histogram Intersection, and logarithmic likelihood statistical method. The chi-square statistics is used to measure the difference of two histograms in this paper, and the formula of chi-square statistics is shown as

| | | |
|-------|-------|-------|
| p_7 | p_0 | p_1 |
| p_6 | p_c | p_2 |
| p_5 | p_4 | p_3 |

FIGURE 2: The window of the LBP algorithm.

$$X^2(S, M) = \sqrt{\sum_i \frac{(S_i - M_i)^2}{S_i + M_i}}. \quad (3)$$

In formula (3), S and M are two different feature vectors and S_i and M_i are the values of the same location in the different vectors S and M .

Figure 3 shows the specific process of the algorithm that detects whether the surveillance camera is occluded. The process that the background modeling method based on background difference and frame difference and the LBP algorithm are used to detect whether the surveillance camera is occluded is as follows:

- (1) Firstly, a surveillance video which is not occluded is used to extract the reference image. Then, the LBP algorithm is used to dispose the reference image and obtain the texture mapping. Finally, the blocked histogram of the mapping is calculated to construct the feature vectors. The feature vectors will be stored and it will be called the reference feature vectors.
- (2) Firstly, the video sequence which will be detected is loaded and the background modeling method based on background difference and frame difference is used to extract the background image. Then, the LBP algorithm is used to dispose the background image and obtain the mapping. Next, the blocked histogram of the mapping is calculated to obtain the feature vectors. Finally, the chi-square statistic is used to measure the similarity between the feature vector and the reference feature vector. And whether the video is occluded at this moment will be determined according to the similarity. The higher the similarity is, the less likely the video is occluded; and the lower the similarity is, the larger the area of the video is occluded.
- (3) If the last frame of the video has not been read, Step 2 will continue to be performed. Otherwise, according to the continuous number that the video is detected as occlusion, whether the surveillance camera is occluded will be output. The more the continuous number is, the more serious the surveillance camera is occluded.

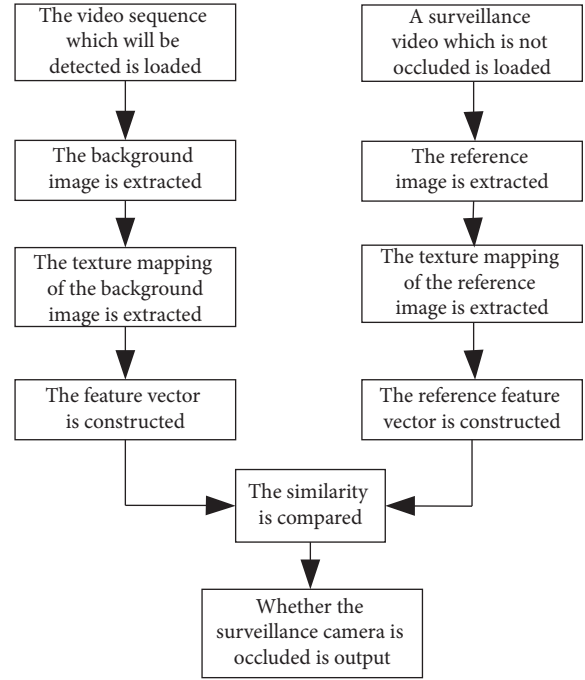


FIGURE 3: The process of occlusion detection for the surveillance camera.

3. Results and Discussion

3.1. The Background Modeling Algorithm Based on Background Difference and Frame Difference. Set Thread1 is 8, Thread2 is 10, and Thread3 is 10. The background image of surveillance video is extracted by using the improved frame difference method in literature [10] and the background modeling method based on background difference and frame difference proposed in this paper. Figure 4 shows the background images extracted from two different videos.

From Figure 4, it can be found that the background modeling algorithm based on background difference and frame difference can extract a better background image with fewer video frames. By comparing the background images extracted from the two videos, it can be found that the background modeling algorithm based on background difference and frame difference has more obvious advantages compared with the original method when the foreground objects whose movement speed is slow exist in the video.

In order to further verify the effectiveness of the background modeling algorithm based on background difference and frame difference, the videos with different occluded areas are shot by using mobile phones. The above two background modeling methods are used to extract the background image of the videos, respectively. Figure 5 shows the results.

From Figure 5 it can be seen that fewer video frames will be used when the background modeling algorithm based on background difference and frame difference is used to

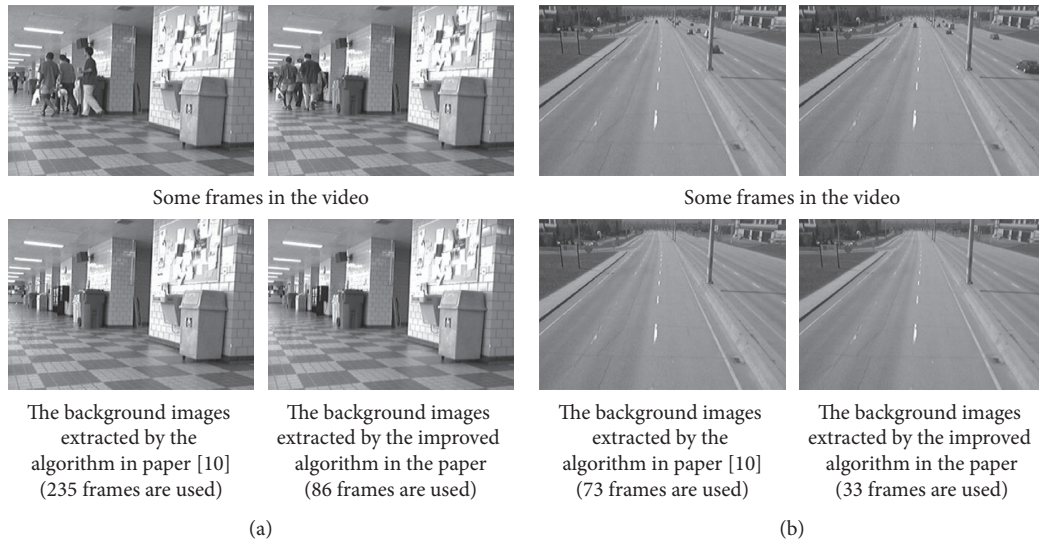


FIGURE 4: The background images extracted by two different algorithms. (a) The background images extracted from the video whose surveillance scene is the hallway. (b) The background images extracted from the video whose surveillance scene is the highway.

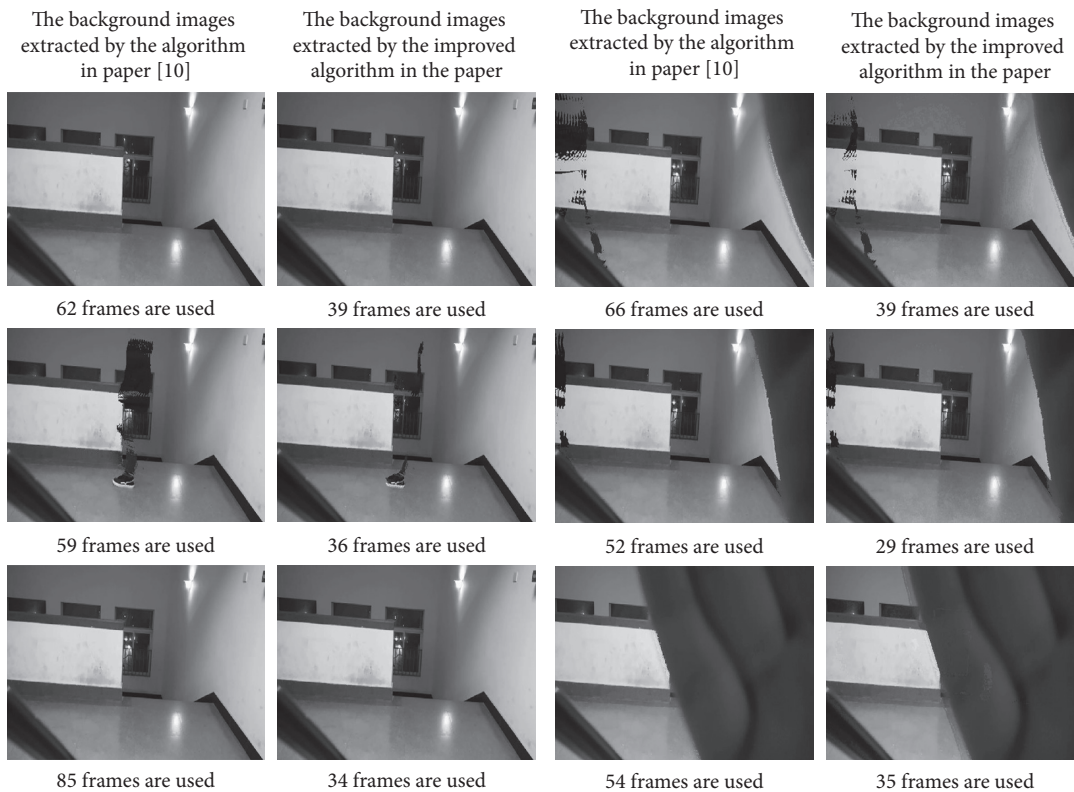


FIGURE 5: The background images extracted from the videos with different occluded areas.

extract the background image for the videos with different occluded areas. By comparing the background images extracted by the two algorithms, it can be found that the background image extracted by the background modeling algorithm based on background difference and frame difference is more effective.

In order to quantitatively compare the improved frame difference method in literature [10] and the background modeling algorithm based on background difference and frame difference, the same videos with different occluded areas are disposed. Each video is 13 seconds long and consists of 400 frames. Using the background modeling



FIGURE 6: When the size of the window is 3×3 , the similarity of the background image with different occluded areas and the reference image. (a) No area occluded 0.14. (b) A small area occluded 0.34. (c) A large area occluded 1.70. (d) A large area occluded 1.83.

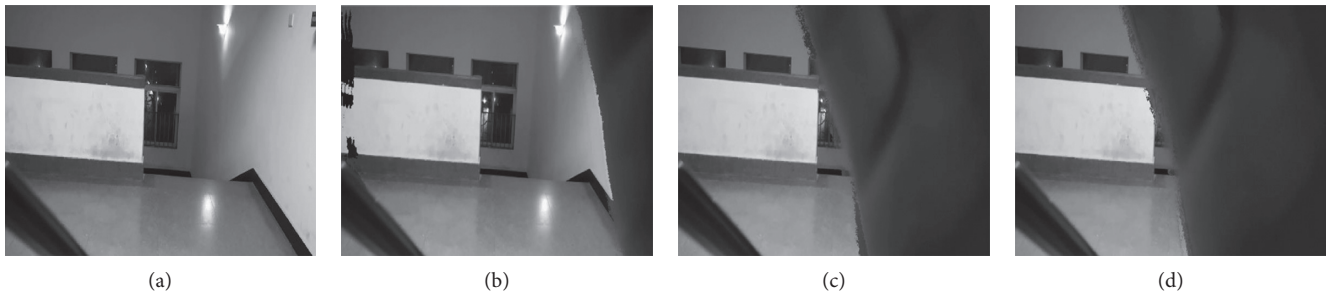


FIGURE 7: When the size of the window is 3×3 , the similarity of the background image with different occluded areas and the reference image. (a) No area occluded 0.16. (b) A small area occluded 1.09. (c) A large area occluded 2.82. (d) A large area occluded 2.89.

algorithm based on background difference and frame difference can extract 87 background images and the success rate of background modeling is 82.76%. Using the improved frame difference method in literature [10] only can extract 48 background images and the success rate of background modeling is 77.08%. So, the background modeling algorithm based on background difference and frame difference is better than the improved frame difference method in literature [10].

3.2. The Algorithm of Occlusion Detection for the Surveillance Camera. The similarity between the background images with different occluded areas and the reference image is calculated by the original LBP algorithm, and the results are shown in Figure 6. Because the occluded areas are a white wall and there is little texture information in it, the texture information in the background image will not change significantly when a small area of the white wall is occluded. For example, the similarity between the background image and the reference image is 0.34 when a small area of the white wall is occluded in the background image. The similarity between the background image which is not occluded and the reference image is 0.14. So, whether the surveillance camera is occluded cannot be easily determined according to the similarity. Only when the similarity gradually increases with the occluded area increase, whether the surveillance camera is occluded can be determined according to the similarity.

After analysis, it is found that this problem can be solved by expanding the window size of the LBP algorithm. Although there is no difference in gray between the white wall

and the white wall, there is a difference in gray between the white wall and the ground. So, the texture features can be extracted according to the difference in gray between the white wall and the ground by adjusting the window size. When the area of the white wall is occluded, the difference in gray between the white wall and the ground will change. So, the texture information extracted will be obviously different. Then, whether the surveillance camera is occluded can be determined according to the similarity between the background image and the reference image. Figure 7 shows the result that the similarity between the background image and the reference image gradually increases with the occluded area increase when the window size is 41×41 in the LBP algorithm. According to the result, the similarity threshold should be preliminarily set at around 1.0. When the similarity between the background image and the reference image is greater than the threshold, it can be determined that the surveillance camera is occluded, and the larger the similarity is, the larger the occluded area will be.

In order to further verify the effectiveness of occlusion detection for the surveillance camera based on a comparison of image similarity, the videos with different occluded areas are disposed. Whether the camera is occluded is determined according to the similarity between the background image and the reference image. Each video is 13 seconds long and consists of 400 frames. When set Thread1 is 8, Thread2 is 10, and Thread3 is 10, about 10 background images can be extracted from each video. It means whether the camera is occluded can be determined every 1.3 seconds. 87 background images are extracted from all videos, and when the similarity threshold is set as 0.8, 0.9, 1.0, 1.1, and 1.2, the

TABLE 1: The result of recognition with different similarity thresholds.

| Similarity threshold | Accuracy rate (%) | False positives rate (%) | False negatives rate (%) |
|----------------------|-------------------|--------------------------|--------------------------|
| 0.8 | 90.80 | 6.90 | 2.30 |
| 0.9 | 89.65 | 6.90 | 3.45 |
| 1.0 | 90.80 | 3.45 | 5.75 |
| 1.1 | 89.65 | 2.30 | 8.05 |
| 1.2 | 88.54 | 1.15 | 10.34 |

accuracy rate of recognition, the false positives rate, and the false negatives rate are shown in Table 1.

From Table 1, it can be seen that the accuracy rate of recognition, the false positives rate, and the false negatives rate are better when the similarity threshold is set as 1.0.

4. Conclusions

Considering previous background modeling methods have the disadvantages that the calculation is complex and constructing the background image takes a long time, the background modeling method based on frame difference is improved in this paper. Combining the background difference, a new background modeling method based on background difference and frame difference is proposed. The simulation results show that fewer video frames are used when the background modeling algorithm based on background difference and frame difference is used to extract the background image, and the background image extracted is better. The above advantages are a good foundation that whether the camera is occluded can be determined by comparing the similarity of the background image and the reference image because the real-time requirements will be satisfied. In the algorithm of occlusion detection for the surveillance camera based on comparison of image similarity, the LBP algorithm is used to compare the similarity between the background image and the reference image. By setting an appropriate similarity threshold, the actual demand can be well met and the application value is very high.

Data Availability

The data used to support the findings of this study have not been made available because they involve the authors' privacy.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Beijing Key Laboratory of Work Safety Intelligent Monitoring, Beijing University of Posts and Telecommunications, Beijing, China.

References

- [1] C. Kuang, *Research on Detection and Classification of Camera Tampering Events*, Harbin Institute of Technology, Harbin, China, 2016, in Chinese.
- [2] M. Qiu, *Research and Achievement of the Diagnoses of the Surveillance Video Image Quality*, East China University of Science and Technology, Shanghai, China, 2014, in Chinese.
- [3] E. Ribnick, S. Atef, O. Masoud, N. Papanikolopoulos, and R. Voyles, "Real-time detection of camera tampering," in *Proceedings of the IEEE International Conference on, 2006 Video and Signal Based Surveillance AVSS'06*, Sydney, Australia, November 2006.
- [4] D.-T. Lin and C.-H. Wu, "Real-time active tampering detection of surveillance camera and implementation on digital signal processor," in *Proceedings of the 2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, Piraeus-Athens, Greece, July 2012.
- [5] H. Yin, X. Jiao, X. Luo, and C. Yi, "Sift-based camera tamper detection for video surveillance," in *Proceedings of the 25th Chinese Control & Decision Conference*, Guiyang, China, May 2013.
- [6] P. Gil-Jimenez, R. Lopez-Sastre, P. Siegmann, J. Acevedo-Rodríguez, and S. Maldonado-Bascón, "Automatic control of video surveillance camera sabotage," in *Proceedings of the IEEE International Conference on the Interplay Between Natural and Artificial Computation*, vol. 4528, pp. 222–231, La Manga del Mar Menor, Spain, June 2007.
- [7] H. Liu and Y. Guo, "A vision-based fall detection algorithm of human in indoor environment," *Proceedings of the SPIE*, vol. 10256, Article ID 1025644, 2017.
- [8] Z. Lian and Z. Wang, "Research on vehicle detection method based on background modeling," *International Journal of Advanced Network, Monitoring and Controls*, vol. 3, no. 2, pp. 6–9, 2018.
- [9] T. Yu, J. Yang, and W. Lu, "Background modeling with extracted dynamic pixels for pumping unit surveillance," *Mathematical Problems in Engineering*, vol. 2018, Article ID 8938673, , 2018.
- [10] X. Kong, *Research on Video Anomaly Detection Method Based on Image Analysis*, Beijing University of Posts and Telecommunications, Beijing, China, 2017, in Chinese.
- [11] Y. Yuan, *Research of Identification for Leaf Occlusion in Surveillance Video*, Wuhan University of Science and Technology, Wuhan, China, 2015.
- [12] L. Shi, X. Wang, and Y. Shen, "Research on 3D face recognition method based on LBP and SVM," *Optik*, vol. 220, Article ID 165157, 2020.

- [13] H. Bejaoui, H. Ghazouani, and W. Barhoumi, "Sparse coding-based representation of LBP difference for 3D/4D facial expression recognition," *Multimedia Tools and Applications*, vol. 78, no. 16, pp. 22773–22796, 2019.
- [14] M. Garg and G. Dhiman, "A novel content-based image retrieval approach for classification using GLCM features and texture fused LBP variants," *Neural Computing & Applications*, vol. 33, pp. 1311–1328, 2020.
- [15] Z. Ye, R. Dong, B. Lin, C. Jin, and Y. Nian, "Hyperspectral image classification based on segmented local binary patterns," *Sensing and Imaging*, vol. 21, no. 2, pp. 556–567, 2020.

Research Article

Distribution Network Topology Identification Based on IEC 61850 Logical Nodes

Yu Chen ¹, Lingyan Sun ¹, Zonghui Wang,¹ and Jinghua Wang²

¹School of Electrical and Electronic Engineering, Shandong University of Technology, Zibo 255000, China

²Shandong Kehui Electric Automation Co., Ltd., Zibo 255000, China

Correspondence should be addressed to Yu Chen; chenyu@sdut.edu.cn

Received 3 October 2020; Revised 23 November 2020; Accepted 6 February 2021; Published 22 February 2021

Academic Editor: Ting Yang

Copyright © 2021 Yu Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Distributed control has good real-time performance and can better meet the control requirements of active distribution networks with a large number of distributed generations. Some distributed applications require real-time feeder topology to achieve control. In this paper, the demand for distributed control applications for feeder real-time topology is analyzed. Based on IEC 61850 modeling method, a new cell topology logic node and a new topology slice node are built to express feeder topology. Using the topology information of smart terminal unit (STU) configuration and the current status information of switchgear, based on the depth-first search, the feeder real-time topology identification can be realized, which meets the application requirements of distributed control. The study case verified the effectiveness of the method.

1. Introduction

Distributed control has good real-time performance and can better meet the control requirements of active distribution networks with a large number of distributed generations. For distributed applications that need to use feeders or local information to make decisions, a real-time feeder topology is needed to achieve control. Taking distributed feeder automation (FA), which is a typical application of distributed control, as an example, it can realize rapid fault location, isolation, and service restoration of feeders, shorten the power outage time of nonfaulty sections to a few seconds, and improve power supply reliability [1, 2].

Distributed FA does not rely on the master station, it only needs the communication between the distribution STUs to make decisions, so it can effectively improve the processing speed of distribution network fault. The implementation of the distributed FA function needs to know the current real-time topology of the feeder [3, 4], especially the location of the tie switch when the power supply is restored. However, due to the operation of fault isolation, load transfer, and network optimization, the location of the tie switch may change. Therefore, for distributed feeder

automation, how to identify the real-time topology of the distribution network is the key problem to be solved [5, 6].

The real-time topology of the distribution network is determined by its static topology combined with the on-off state of switches on the feeder. The purpose of topology identification is to form the current feeder real-time topology according to the switch real-time state for the operation control of the distribution network [7, 8]. At present, there are two modes of distribution network topology processing; one is the master station processing mode and the other is the distributed processing mode. Some DMS functions need the real-time topology of the distribution network, such as centralized FA. The topology identification algorithm mainly includes the tree search method and adjacency matrix method [9, 10]. In the centralized FA mode, the STU on the feeder does not store the feeder topology information. The static topology of feeders is stored in the master station, and the real-time topology is obtained by the topology recognition algorithm. When the topology of distribution lines changes, the complete topology of feeders will be updated in the master station. In this way, the topology information is complete, but it can only be used for centralized processing applications [11]. In order to

complete the control, the distributed control application needs the real-time topology of the feeder, but it generally does not need the complete topology of the feeder. It only needs the topology information in the control domain [12]. In order to meet the needs of distributed control topology information, the research of real-time distribution line topology identification has been carried out in the literature. For the expression and configuration of topology, Zhu and Cong' team configure the adjacency relationship of adjacent switches on feeders based on the user-defined format to realize the feeder automation function [13, 14]. Zhu' team describes the distribution network topology based on the Process and Line model of IEC 61850 system configuration language (SCL) and proposes a distributed topology processing method based on Graph Segmentation and realizes the topology identification of distribution network by communication of STUs [15, 16]. The topology description method based on SCL needs to generate a topology configuration file and store it in the STUs. The real-time topology is obtained by transferring and exchanging the topology configuration file. Except for the topology configuration, other functions are achieved by logical nodes. Fan' team proposes a method, which uses the STU local topology matrix, and realizes topology recognition after exchanging information between STUs [17].

In this paper, the IEC 61850 method is used to model logical node (LN) RTCN and RTPM to express the topology in the feeder. Based on the logical nodes, the topology recognition algorithm is studied to complete the topology search of feeders to meet the application requirements of distributed control applications. Compared with the method of describing distribution network topology based on SCL, the method of describing topology information based on logical nodes, and the method of configuration and acquisition of local topology are consistent with other functional logical nodes, which is more convenient for application and promotion of distributed control application.

2. Distributed Control Application Topology

2.1. Distributed Control. Distributed control application deploys functions to STU and uses STU's communication with each other to exchange detection and control information to achieve corresponding functions. Taking distributed FA as an example, after STU on the line switch detects the fault information, they exchange information and decide to isolate the fault section. After the fault section is isolated successfully, the nonfault upstream section is restored by closing the main breaker. If there is a tie switch in the nonfault downstream section of the fault section, it is restored by closing the tie switch.

Taking Figure 1 as an example, CB is the circuit breaker, S1, S2, S3, and S5 are sectional switches, and S4 are tie switches. STU is installed at the main breaker, section switch, and tie switch, and STU communicates through a peer-to-peer communication network.

In the application of distributed FA, it is necessary to know the topology relationship of each device. When the fault isolation of distributed FA is completed, the power

supply of the nonfault section needs to be restored. The system needs to know the topology of the fault line downstream to determine the appropriate power supply recovery path.

2.2. Static Topology. Network static topology refers to the static adjacent relationship between the distribution equipment and the line. The change of the adjacent relationship between the distribution equipment will lead to the change of the static topology. The construction of distribution lines, such as line extension, switch increase and decrease, and new equipment put into operation, will change the static topology of the network.

2.3. Application Topology. Application topology refers to the real-time adjacent relationship of related devices in the control domain when a specific function is realized. The real-time topology of the feeder is determined by the static topology combined with the switching state of the switch. Taking the distributed FA as an example, the real-time topology generally refers to the real-time connection relationship of the medium-voltage distribution feeder starting from the substation bus, and the endpoint is the load, the opposite substation bus, the distributed generation, and the tie switch. When fault location, isolation, and recovery, it is necessary to know the upstream and downstream connection of real-time STU. In this paper, based on the new logical node to express the static topology of the feeder, combined with the current switch state information, the search algorithm is used to obtain the real-time application topology of the feeder.

The change of switch state will cause the change of application topology, as shown in Figure 1. Before the fault, the connection relationship of the distribution network is CB1-S1-S2-S3 and CB2-S5-S4. After the failure, due to the switch operation, the application topology changes, as shown in Figure 2, and its connection relationship becomes CB1-S1, CB2-S5-S4.

3. Distribution Network Topology Representation

3.1. Existing Distribution Network Topology Model. The topological model of the network is the basis of all network analysis applications. The substation configuration description language SCL defined in IEC 61850-6 Ed2.1 can describe the feeder topology, intelligent electronic device (IED) information model, and communication service of the system. The work by Zhu' team [18], IEC 61850-6 Ed2.1, describes the distribution network topology through new Process and Line elements. In the SCL model description of IEC 61850 Ed2.1, the main line and branch line in the distribution network are all represented by the Line element, and the medium-voltage/low voltage distribution substation and switching station in the distribution network are described by Substation. When the network contains both Substation and Line, the process container needs to be used on the upper layer to represent the local network of a system. After the topological model of feeders described by SCL is

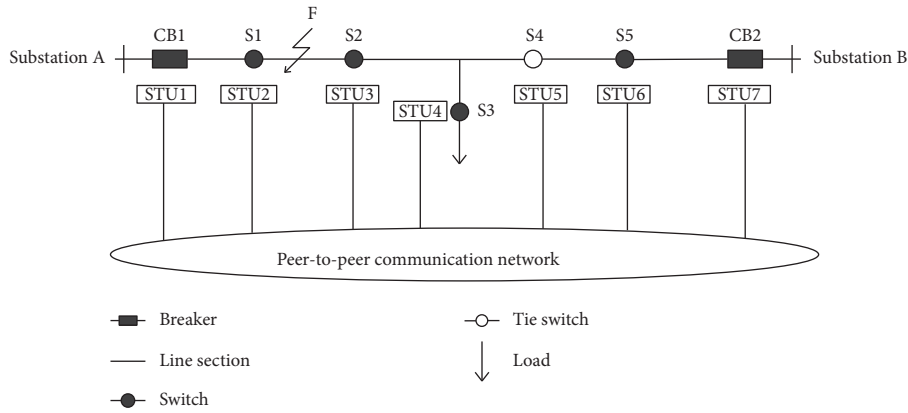


FIGURE 1: Typical distribution lines.

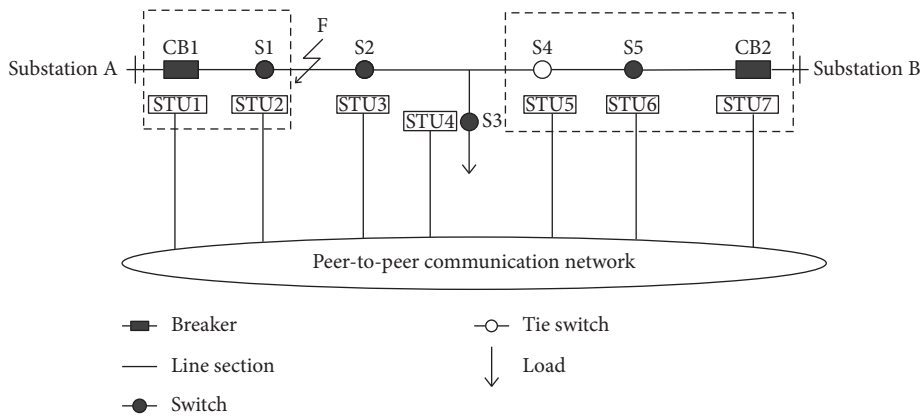


FIGURE 2: Typical distribution lines after fault.

established, the topological model needs to be configured in the IED of feeders. There are two configuration modes: the first is to divide the complete topology model of the feeder in the master station and then send it to the corresponding IED on the feeder to complete the configuration, the second is to use the SCL file on the IED to describe the local topology, and the IED uploads the master station to form the complete topology of the feeder in the master station.

3.2. Solution of Topology Description Based on Topology Node. For distributed FA applications, STU needs to know the current topology connection relationship to perform fault location and isolation and need to define the complete topology of the feeder to determine the recovery power supply path.

To complete the distributed FA application, the detailed feeder complete topology should be configured in the STU, which needs large storage space and is difficult to deal with when using, so it is necessary to study how to effectively simplify the expression of topology.

When using the topology relationship, distributed FA is mainly concerned with the connection relationship of conductive equipment and does not need the length of wire and other detailed electrical parameters.

In the topological description, we can mode the wire and regard it as a logical connecting line. After the wire is modeled as a logical connecting line, the configuration of the STU can be simplified, and the implementation of distributed FA applications can be better supported.

Based on this, this paper studies the topology description and uses IEC 61850 modeling method to create a new logical node to represent the topology structure.

3.2.1. Cell Topology Logical Node. Based on the modeling rule of IEC 61850, a new logical node of cell topology is built, which is the smallest logical unit of feeder topology. Each cell topology logic node represents the connection relationship between the conductive equipment on the feeder and its adjacent conductive equipment.

In the equipment model in CIM, equipment models are divided into conductors, transformer windings, loads, connectors, equivalent power supplies, regulating equipment, and switches. Referring to CIM's equipment modeling, this paper divides the equipment on the distribution line into the equipment with breaking capacity and the equipment without breaking capacity. The equipment with breaking capacity is uniformly defined as switch type, and the equipment type is also defined for the bus and

transformer, which do not have breaking capacity but affect the boundary judgment of topology identification.

The smallest unit of the feeder is represented by the logical nodes of cell topology, and the local feeder topology can be represented by multiple logical nodes of cell topology, and then the local topology is formed by the integration of the logical nodes of cell topology. When the network changes, only the data of the cell topology logical node corresponding to the changed device needs to be updated.

3.2.2. Topology Slice Node. Because STU is not installed on every switch on the feeder, a container is needed to store a local area topology of multiple switches. The topology slice node represents the local area topology of feeders. The local area topology is composed of multiple cell topology logical nodes, so the topology slice node contains all the cell topology logical nodes that make up the local area topology. The complete topology of feeders is composed of several local area topologies. When the adjacency relationship of local area topology is expressed clearly, multiple local area topologies can be combined into a complete topology of the feeder.

When describing the topological relationship of feeders, all conductive devices have unique names. In the description of the topology slice, the boundary of the topology slice needs to be determinate. The topological slice is bounded by bus bars, transformers, and tie switches. The cell topology logic node search in the topology slice is limited to the boundary of the topology slice. The topology slice generally corresponds to the control area of the STU.

4. Topological Logical Node Modeling

4.1. Cell Topology LN Modeling. According to the requirements of distributed FA application for topology, the cell topology logic node needs to express (1) conductive equipment on the feeder, (2) type of conductive equipment, (3) conductive equipment adjacent to the equipment, and (4) number of adjacent conductive equipment. Based on this, a new cell topology logical node RTCN is created, where R represents that the logical node belongs to the protection related function node group, T is the abbreviation of topology, and CN represents the connection node. The main data objects of the cell topology logical nodes are shown in Table 1. M/O/C means required/optional/condition required.

The CeName attribute in the cell topology logical node represents the name of the conductive device. PTRType refers to the type of conductive equipment, which can be distinguished according to whether it has breaking capacity. The equipment with breaking capacity on the line is a switch, which is represented by constant 1 in PTRType attribute, while the equipment without breaking capacity is represented by substation bus, switch station bus, and transformer, which is represented by constant 2, 3, and 4 in PTRType attribute. The AdjRTCNum attribute represents other cell topology logical nodes connected to the cell topology logical node. In this data object, there can be multiple cell

topology logical nodes adjacent to it. Multiple cell topology logical nodes form an array, and the attribute type is a character string. The AdjRTCNum attribute indicates the number of adjacent cell topology logical nodes.

4.2. Topology Slice LN Modeling. On the basis of the logical node of the cell topology, a new topology slice node (RTPM) is needed to describe the local area topology of the feeder. The topology slice node (RTPM) also belongs to the protection related function node group. The topology slice node needs to express (1) the cell topology logic node covered by this topology slice, (2) other topology slices adjacent to this topology slice node, (3) the number of adjacent topology slices. The data objects of the topology slice node are shown in Table 2.

The function of the topology slice node is to describe the connection relationship between the topology slices and include all the cell topology logical nodes in the topology slice. The following is a description of the main data objects of the topology slice node:

- (1) AdjRTPM attribute refers to the adjacency topological slices of this topological slice. All adjacency topological slices form an array, and its attribute type is a string.
- (2) The AdjRTPMNum attribute represents the number of adjacent topologies. The attribute type is a numeric constant.
- (3) The RTPMCovRTCNum attribute represents the cell topology logical nodes covered by this topology slice. All the cell topology logical nodes form an array, and the attribute type is a string.

As shown in Figure 3, the relationship between the cell topology logical node and the topology slice node and the relationship between the local topology stored in the corresponding STU are shown. The cell topology logical nodes RTCN1 and RTCN2 belong to the topology slice RTPM1, and the topology slice RTPM1 is stored in STU1.

Taking the feeder in Figure 3 as an example, the topology of the unit topology logical node and the topology slice node is described as follows:

- (1) The AdjRTPM attribute of the topology chip RTPM1 stored in STU1 indicates that the adjacency topology slice node is RTPM2, AdjRTPMNum indicates that the number of adjacency topology slices is 1, and the RTPMCovRTCNum attribute indicates that the RTPM1 contains the unit topology logic nodes RTCN1 and RTCN2. RTPM1 data block diagram is shown in Figure 4.
- (2) RTCN3 included in RTPM2 indicates the switch K1 on the line. The AdjRTCNum attribute in RTCN3 indicates that the adjacent cell topological are RTCN2, RTCN4, and RTCN6. Its PTRType attribute is the constant 1, which represents a switch on the line. The CeName attribute indicates that the specific equipment name of the distribution line is 10 kV**line\$K1. RTCN3 data block diagram is shown in Figure 5.

TABLE 1: Main data objects of RTCN.

| Attribute name | Attribute types | Explanation | M/O/C |
|----------------|---------------------------------|------------------------------|-------|
| CeName | VISIBLE STRING | Name of conductive equipment | M |
| PTRType | INS | Type of conductive equipment | M |
| AdjRTCN | ARRAY[0..Num] of VISIBLE STRING | Adjacency RTCN | M |
| AdjRTCNum | INS | Number of adjacent RTCN | M |

1- Switch
 2- Substation bus
 3- Bus of switching station
 4- Transformer
 5- other

TABLE 2: Main data objects of RTPM.

| Attribute name | Attribute types | Explanation | M/O/C |
|----------------|---------------------------------|-------------------------|-------|
| AdjRTPM | ARRAY[0..Num] of VISIBLE STRING | Adjacency RTPM | M |
| AdjRTPMNum | INS | Number of adjacent RTPM | M |
| RTPMCovRTCN | ARRAY[0..Num] of VISIBLE STRING | RTCN included in RTPM | M |

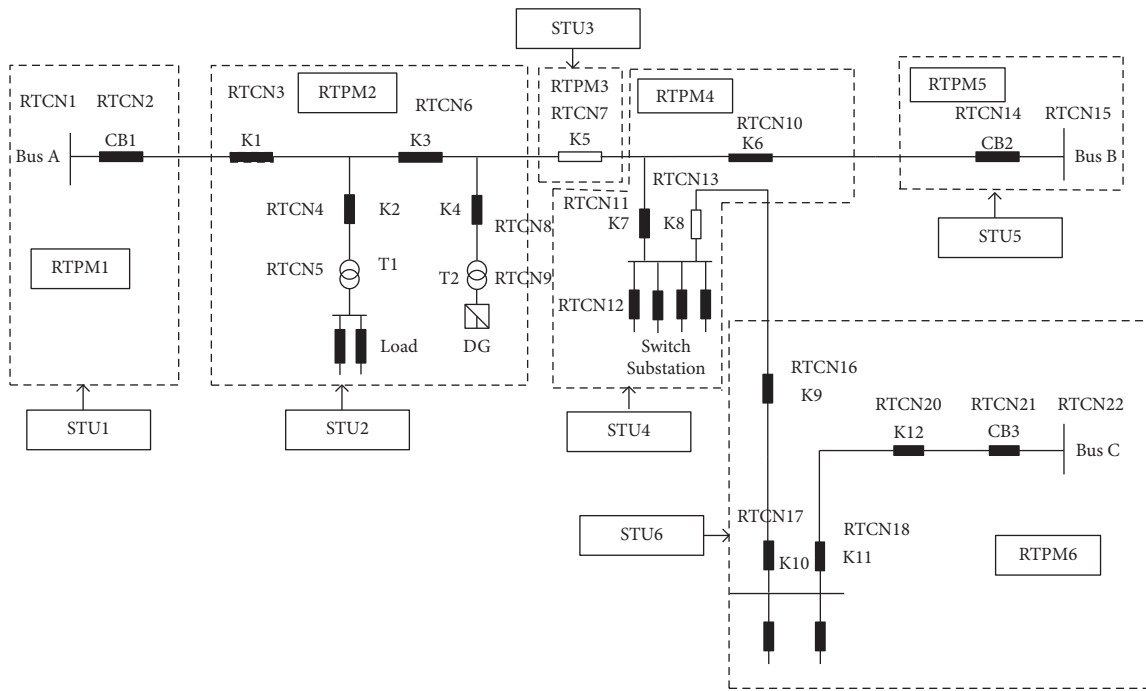


FIGURE 3: Schematic of distribution network topology.

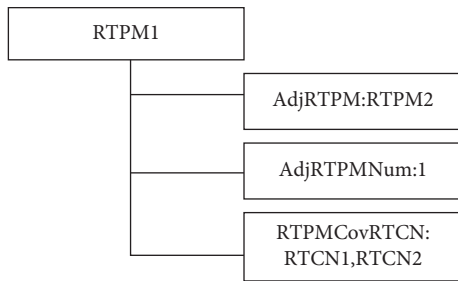


FIGURE 4: Topological slice nodes RTPM1.

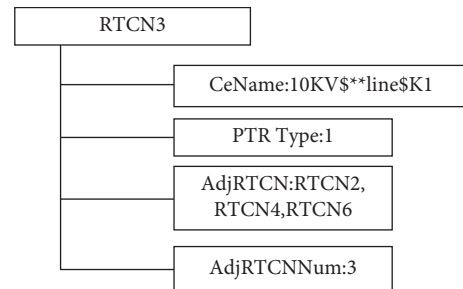


FIGURE 5: Logical node RTCN3.

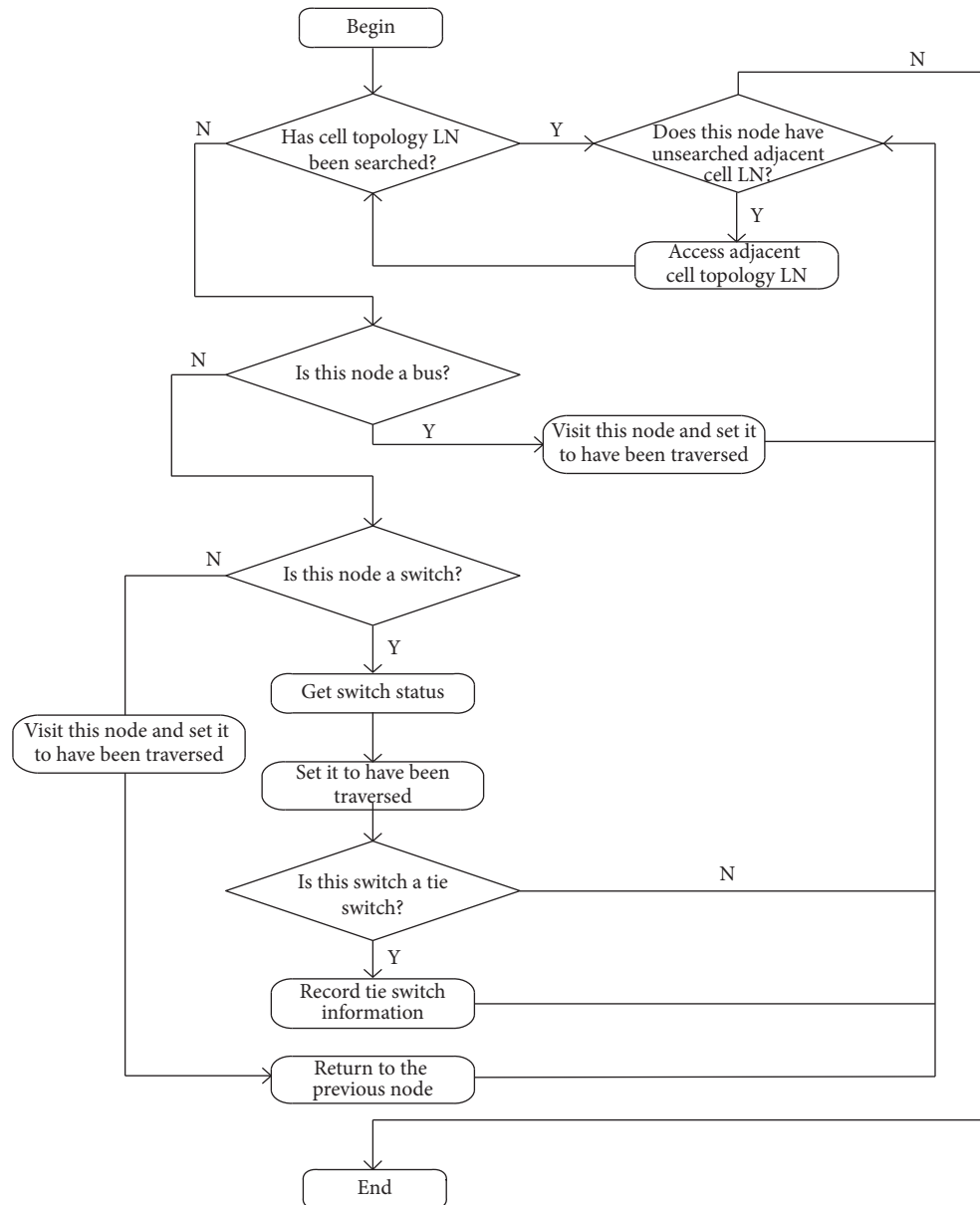


FIGURE 6: Algorithm flowchart.

5. Topology Recognition Based on Topology LNs

When equipment is installed or reduced in the distribution line, or the state of the tie switch changes, the corresponding STU notifies the distributed FA application master STU to initiate a topology search and update the application topology. Reading the local static topology configured in each STU and searching according to the current switch status can obtain the application topology. The GetDataValues service of IEC 61850 ACSI can complete the transmission of topology data.

Topology recognition is mainly two steps: first, search in the local topology configured by STU itself. When an adjacent cell topology LN is not in the current topology slice, the adjacent topology slice needs to be obtained. The STU

initiates communication with the STU corresponding to the adjacent topology slice node, obtains the cell topology LNs of the adjacent topology slice, and then searches and connects multiple local area topologies to form a complete application topology of the feeder. The flow chart of the topology search algorithm is shown in Figure 6.

The process of connecting the logical nodes of the cell topology to form a topological slice is the process of identifying the local area topology of the line. In order to realize the real-time topological identification of lines, the first step is to traverse the graph. Starting from a given cell topology logical node in a connected graph, all cell topology logical nodes in the graph are accessed, and each cell topology logical node is accessed only once. There are two ways to traverse the connected graph: depth-first search (DFS) and breadth first search (BFS). In this paper, we use the static

topology information stored in STU and the switch state on the feeder to identify the local area topology in the topology slice.

The feeder topology information represented by the cell topology LNs and the topology slice LNs is saved in the STU on the line, and each STU only stores the topology information related to the control domain of this STU. It is necessary to carry out real-time topology search across topology slices to transfer the local area topology information saved by STU. The complete topology search of feeders across the topology slice needs to select a master STU, and the master STU obtains the topology information saved by other STU to form the complete topology information of feeders.

The LNs of cell topology and depth-first algorithm are combined to search the feeder topology, and the bus is the starting point of topology search. Taking the distribution line shown in Figure 3 as an example, when the distribution network topology changes, STU1 is selected as the master STU to initiate the topology search. The search steps are as follows:

- (1) Traverse from RTCN1, which represents bus A. Access RTCN1 and mark this node as traversed. The AdjRTCN element in RTCN1 points to RTCN2, a neighboring cell topology logical node that is not traversed.
- (2) Access the cell topology logical node RTCN2, which represents the outgoing breaker CB1. Get the switch status of the outgoing breaker CB1, and mark this node as traversed. RTCN1 in the AdjRTCN element of RTCN2 has been traversed, while RTCN3 has not. There is no information of RTCN3 in the topology slice RTPM1, so STU1 communicates with STU2 corresponding to the adjacent topology slice RTPM2 to obtain the cell logic node information in the topology chip RTPM2, and carry out the next topology search across the topology slice.
- (3) Access the cell topology logical node RTCN3. RTCN3 represents the switch K1, obtains the switch state of K1, and marks this node as traversed, and the topology search continues. RTCN2 in AdjRTCN element of RTCN3 has been traversed, while RTCN4 and RTCN6 have not. You can continue to search for RTCN4 and RTCN5 in this way.
- (4) Access the cell topology logic node RTCN6. RTCN6 represents switch K3, obtains the switch state of K3, and marks this node as traversed. Switch K3 is a tie switch. RTCN3 and RTCN4 in the AdjRTCN element of RTCN6 have been traversed, while RTCN7 has not.

The search procedure of the cell topology logical node in RTPM3 is the same as the above steps. After the cross topology search is completed, the topology slices RTPM1, RTPM2, RTPM3, RTPM4, RTPM5, and RTPM6 are connected to form the whole feeder topology.

So far, the complete topology identification of feeders is completed. When the switch operates or the feeder structure changes, only the changed topology piece information needs to be updated to complete the real-time topology update.

6. Summary

Distributed control has perfect performance and fast response speed, which can effectively improve the safety of the distribution network and improve the power supply quality of the distribution network. Based on the IEC 61850 modeling technology, this paper builds new cell topology logical nodes and topology slice logical nodes based on the analysis of distributed application topology requirements and uses logical nodes to express the distribution line's topology. This method does not need to name each section of the line, simplifies the description of the topological structure, and can better support the realization of distributed control.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work was supported by the Key Project of Smart Grid Technology and Equipment of National Key Research and Development Plan of China (No. 2016YFB0900600) and Technology Projects of State Grid Corporation of China (No. 52094017000W).

References

- [1] J. Zhou, L. Wang, M. Liu et al., "Research on Quick Distributed Feeder Automation for Fast Fault Isolation/Self-Healing in Distribution Network," in *Proceedings of the 2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, pp. 202–206, Chengdu, China, May 2019.
- [2] Y. Li, "An Improved Fast Self-Healing Feeder Automation System," in *Proceedings of the IEEE Advanced Information Technology, Electronic and Automation Control Conference*, pp. 2428–2432, Xi'an, China, May 2018.
- [3] L. Wang, J. Zhou, M. Liu et al., "Research on Application of New Intelligent Distributed Feeder Automation," in *Proceedings of the 2019 IEEE 3rd International Electrical and Energy Conference (CIEEC)*, pp. 182–186, Beijing, China, September 2019.
- [4] G. Zhabelova and V. Vyatkin, "Multiagent smart Grid automation architecture based on IEC 61850/61499 intelligent logical nodes," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 5, pp. 2351–2362, 2012.
- [5] N. Kashyap, C.-W. Yang, S. Sierla, and P. G. Flikkema, "Automated fault location and isolation in distribution grids with distributed control and unreliable communication," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2612–2619, 2015.

- [6] J. Liu, B. Yun, Q. Cui et al., "A distributed intelligent feeder automation system with fast self-healing performance," *Automation of Electric Power Systems*, vol. 34, no. 10, pp. 62–66, 2010.
- [7] Z. Zhu, Z. Jin, J. Chen et al., "Topology Self-Identification and Adaptive Operation Method of Distribution Network Protection and Self-Healing System," in *Proceedings of the 2018 International Conference on Power System Technology (POWERCON)*, pp. 3087–3092, Guangzhou, China, November 2018.
- [8] Z. Zhu, B. Xu, G. Han et al., "Study on the Feeder Topology Modeling and IED Configured Methods for IEC61850," in *Proceedings of the 2014 China International Conference on Electricity Distribution (CICED)*, pp. 1482–1487, Shenzhen, China, September 2014.
- [9] J. Ying, L. Chen, M. Liu et al., "Research on Check Method of Feeder Topology Model for Distribution Main Station," in *Proceedings of the 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pp. 1–5, Beijing, China, October 2018.
- [10] M. Zhang, P. Shen, W. Ji et al., "Feeder Topology Representation Method Based on Multiway Tree," in *Proceedings of the 2016 IEEE International Conference on Power and Renewable Energy (ICPRE)*, pp. 451–455, Shanghai, China, October 2016.
- [11] H. Liu, L. Mu, J. Su et al., "Centralized and intelligent control mode of feeder automation," *Power System Technology*, vol. 31, no. 23, pp. 17–21, 2007.
- [12] R. Chen, J. Lu, M. Liu et al., "Distribution Network Topology Model Generation Method for Distributed Feeder Automation," in *Proceedings of the 2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, pp. 1057–1062, Chengdu, China, May 2019.
- [13] G. Zhu, P. Shen, Y. Wang et al., "Dynamic identification method of feeder topology for distributed feeder automation based on topological slices," *Power System Protection and Control*, vol. 46, no. 14, pp. 152–157, 2018.
- [14] W. Cong, Y. Zheng, Z. Zang et al., "Distributed storage and management method for topology information of smart distribution network," *Automation of Electric Power Systems*, vol. 41, no. 13, pp. 111–117, 2017.
- [15] Z. Zhu, B. Xu, C. Guise, and G. Han, "Distributed topology processing solution for distributed controls in distribution automation systems," *IET Generation, Transmission & Distribution*, vol. 11, no. 3, pp. 776–784, 2017.
- [16] Z. Zhu, *Key Technologies for the Application of IEC 61850 to Distributed Controls in Smart Distribution Grids*, Shandong university, Jinan, China, 2018.
- [17] K. Fan, B. Xu, J. Dong et al., "Identification method for feeder topology based on successive polling of smart terminal unit," *Automation of Electric Power System*, vol. 39, no. 11, pp. 180–186, 2015.
- [18] Z. Zhu, B. Xu, Y. IP. Tony et al., "IEC 61850 based models for distributed feeder automation system," *Automation of Electric Power Systems*, vol. 42, no. 23, pp. 148–154, 2018.

Research Article

Research on Distributed Feeder Automation Communication Based on XMPP and GOOSE

Lingyan Sun ¹, Yu Chen ¹, Chuiyue Kong,¹ and Jinghua Wang²

¹School of Electrical and Electronic Engineering, Shandong University of Technology, Zibo 255000, China

²Shandong Kehui Electric Automation Co., Ltd., Zibo 255000, China

Correspondence should be addressed to Lingyan Sun; 591824299@qq.com and Yu Chen; chenyu@sdut.edu.cn

Received 7 October 2020; Revised 14 December 2020; Accepted 28 January 2021; Published 8 February 2021

Academic Editor: Ting Yang

Copyright © 2021 Lingyan Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the application process of distributed feeder automation (FA) that is based on peer-to-peer exchange of measurement and control data between smart terminal units (STUs), there is an urgent need for standardized communication interaction and necessary security protection. This paper proposes an IEC 61850 communication mapping scheme using built-in secure extensible messaging and presence protocol (XMPP) and the generic object oriented substation event based on the user datagram protocol (GOOSE over UDP) and a security protection scheme based on hash to obtain random subsets (HORS); one-time signature algorithm is used to ensure the communication safety of GOOSE messages. The agent-based distributed FA test system is developed with the STUs. The test results show the scheme can meet the requirements of the quick distributed feeder automation.

1. Introduction

The structure of the distribution line is complex and the failure rate is high. At the same time, the requirements for power supply quality and reliability from users become stringent. The distributed feeder automation can effectively speed up the failure processing speed and improve the power supply reliability [1, 2]. The technology of distributed FA based on the peer-to-peer exchange of measurement and control data between STUs has attracted wide attention due to its comprehensive utilization of information, fast local control speed, and perfect performance. However, there are several issues to be solved in the existing FA system: (1) the private information model and communication mechanism are adopted in the information communication of STUs of various manufacturers, which lack the support of standard information model and communication mechanism, resulting in the failure to realize the interoperability of different STUs; (2) the communication between STUs lacks the unified service scheme and necessary security protection technology.

Distributed FA control is to realize fault handling through peer-to-peer communication between STUs, which requires a unified information model, communication

mapping and reliable communication network. In terms of information model, at present, the information model of distribution network is mainly based on IEC 61850. In the work by Ling's team [3], the FA controller is used to realize the control of the terminal and the expansion and control of the logical node to complete the distributed FA control; chen's team [4] proposed a peer-to-peer communication data exchange method for GOOSE services and established proprietary logical node and smart distributed FA model. In terms of information model, new or expanded logical nodes are often used to meet the requirements of distributed FA. In terms of communication network, IP communication network is generally used and fiber-optic communication has become the first choice of distribution communication network due to its reliable performance and strong anti-interference ability. Data transmission is mostly realized by the method of communication mapping. At present, the research on communication transmission protocol mainly includes IEC 60870-5-101/104, MMS, Web Services, and GOOSE. Among them, MMS is mainly used in substation and Web Services cannot meet the requirements of real-time performance and security of distribution network, so there are few applications of distribution network. IEC 60870-5-

101/104 is the most widely used communication mapping protocol in distribution network. In order to ensure the security of communication information, when IEC 60870-5-101/104 protocol is applied in distribution network, China state grid requires encryption when control command is issued, but it only solves the issue of longitudinal security from master station to terminal and does not solve the lateral security from terminal to terminal. In order to solve the issue of communication security, the working group of IEC TC57 proposed XMPP communication mapping and is developing corresponding standards; Hussain's team [5] studied service mapping scheme for IEC 61850-based XMPP communication; Wang's team [6] optimized and improved the mapping scheme of XMPP and verified the simplicity and efficiency of the scheme to achieve interoperability among devices with limited resources in the Internet of Things. Cho's team [7] has built an XMPP platform based on the IEC 61850 of the Internet of Things, which can effectively monitor DER; Hou's team [8] studied the real-time communication of XMPP and verified that the communication delay time mapped from IEC 61850 to XMPP can meet the real-time performance requirements of master station and STUs in distribution automation, and the communication delay time between STUs can meet the real-time performance requirements of slow distributed FA but not the quick distributed FA. For the fast transmission technologies of real-time control data, GOOSE transmission mechanism is often used. In the work by Chen's team [9], the mapping scheme of existing GOOSE is introduced in detail, and the optimized GOOSE mapping method based on TCP protocol is proposed. In the work by Fan's team [10], the existing GOOSE mapping is analyzed based on the requirements of distributed control communication, and the mapping scheme of GOOSE over UDP is proposed. Chen and Fan et al. have verified through experiments that the real-time performance of the GOOSE mechanism can meet the requirements of distributed control [9, 10]. The above studies have solved the cross-communication network issue when GOOSE transmission is used in the distribution network, but none of them considers the security protection of GOOSE transmission.

In order to realize the interoperability between the STUs of the distributed FA in the distribution network and effectively solve the issue about communication security, this paper studies the solution based on the combination of XMPP and GOOSE over UDP, the XMPP protocol mapping is used to realize the transmission of conventional data, and GOOSE over UDP is used to realize the transmission of real-time control data (such as switch action and protection trip). A one-time signature algorithm is used to solve the security protection issue of the GOOSE mechanism, and the real-time performance of the proposed transmission scheme is tested on the constructed platform.

2. Distributed FA

The distributed FA system is composed of master station of distribution automation system, STUs, and peer-to-peer communication network. Its main functions are as follows:

(1) when the system is in normal operation, the STUs monitors the corresponding primary switchgear status information and reports it to the master station; (2) when a fault occurs on the system, peer-to-peer real-time interactive data between STUs realize fault location, isolation, and service restoration, that is, FLISR function, and report the processing results to the master station.

When a short-circuit fault occurs on the distribution lines, the outlet circuit breaker and related STUs detect the fault current. The circuit breaker trips to re-move the fault, and the STU that detects the fault current starts the FA function and judge the fault section according to whether there is fault current flowing through adjacent switches. In Figure 1, because STU0 and STU1, respectively, detect that there is fault current flowing at CB1 and switch S1, it is judged that the fault does not occur in the adjacent section of CB1. STU2 detects that there is no fault current at switch S2 of the switch adjacent to switch S1 and judges that the fault occurs in the section where K1 point is located. After determining the fault section, STU participating in decision control runs FLISR algorithm to generate fault isolation and recovery scheme. STU1 and STU2 execute the command to disconnect switch S1 and switch S2, respectively, isolate the fault section, and send the confirmation message. STU3 and STU0 execute command to close contact switch S3 and circuit breaker CB1 successively to re-store power supply.

If the distribution lines include distributed energy resource (DER), as the access of DER changes the structure of the distribution network and changes in electrical quantities, it is necessary to locate the fault based on the comparison of the magnitude of the fault current or the phase comparison [11].

According to the number of STU involved in decision-making control, the implementation mode of distributed FA can be divided into cooperative mode distributed FA and agent mode distributed FA [8].

2.1. Cooperative Mode Distributed FA. Cooperative mode distributed FA refers to two or more STUs to jointly participate in decision-making to realize the function of distributed FLISR. When a fault occurs on distribution lines, STU which detects fault information of field switch starts FA function, exchanges information with adjacent STU and runs FLISR algorithm, makes logical judgments independently, and determines the fault section. After generating the fault isolation recovery scheme, each STU sends the sequence control command locally, and the corresponding switch executes the action to realize FLISR operation.

2.2. Agent Mode Distributed FA. The agent mode distributed FA is a decision-making control that a designated STU completes the FLISR function. Generally, the STU at the outlet power switch of the substation is selected as the agent STU by taking the feeder as the unit, and the other STU are collectively referred to as the slave STU. Each STU in the ring network transmits the detected information to the corresponding agent STU. The agent STU initiates the logic of fault handling and decides the switch action and transmits the control command to the corresponding slave STU. In

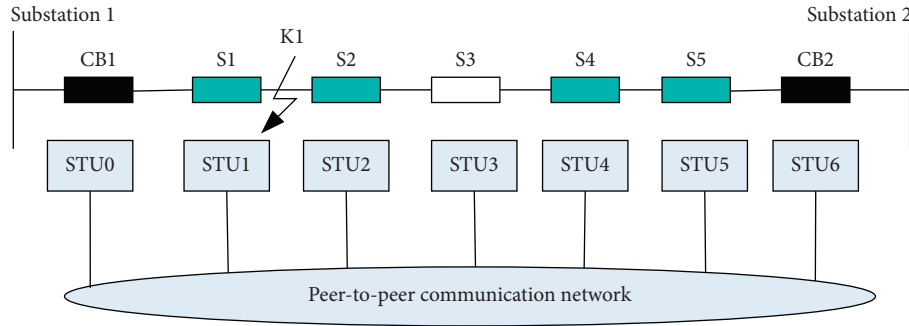


FIGURE 1: A distributed FA system for an open loop overhead line.

this mode, the principle of distributed FA is similar to centralized FA [8], in which the agent STU is equivalent to the substation of distribution network.

2.3. Real-Time Performance Requirements of Distributed FA. Distributed FA can effectively speed up the processing speed of distribution lines faults and reduce the outage time. According to different load types and communication conditions, it can be divided into quick distributed FA and slow distributed FA. The China national standard GB/T 35732-2017, Technical Specifications of Intelligent Remote Terminal Unit of Distribution Automation [12] specifies the information interaction and fault processing time, as shown in Table 1.

2.4. Communication Security Requirements of Distributed FA. In recent years, network security incidents occur frequently. In 2015, Ukraine, and in 2016, Israel, suffered hacker attacks on the power system, resulting in power outage. This indicates that the risk of grid network and information security exists for a long time and needs to be effectively protected.

In the distributed FA system, the data interaction object is mainly the STU and the master station. The STUs are distributed outdoors and scattered in a broad area in distribution automation system. Most of the environments are unattended and vulnerable to attack. The content of data interaction involves real-time measured current and voltage information, fault indication, switch position, control commands, etc. and its operation object is directly oriented to sectionalising switches. If the STUs are attacked or the interactive data are leaked or tampered and the wrong instructions are conveyed, the circuit outage and other accidents will occur directly.

Technical report IEC 61850 90-5 is used in the wide area phase angle measurement application in combination with communication security standard IEC 62351, and the method of establishing key distribution center (KDC) signature authentication is used for security protection. This method is more suitable for agent mode distributed FA, when applied to cooperative mode distributed FA, the number of key required is large, and the management is complex. In addition, The STU as the KDC is limited due to its computational ability, the key cannot be too long, and the security protection capability is limited.

At present, for the security protection of distribution network, China state grid clearly states that the security protection must be done according to the following requirements: the master station in distribution automation system should meet the one-way authentication function of nonsymmetric encryption key technology, and the STUs should have the function of authenticating the digital signature of the master station, but it only involves the security protection between the STUs and the master station; it does not require the security issues between the STUs. At the same time, it needs to strengthen the safety monitoring and management of the STUs and other equipment. Electrical Internet of Things also puts forward requirements for grid security: eliminate the weak links in the grid, use new technologies or new methods to improve the security protection of important equipment and time periods in the grid, and strengthen the security prevention and control of important information transmission to prevent the impact of “network attack” on the grid.

In the current IEC 61850 communication protocol, XMPP can support various kinds of security encryption algorithms, so in this paper, XMPP mapping communication is used for information model and measurement data, and GOOSE over UDP is used for real-time control command; at the same time, security protection is added to it.

3. Communication Mapping of XMPP in Distributed FA

3.1. XMPP Working Mechanism. XMPP is an open-source communication protocol for real-time communication. It is based on extensible markup language (XML), which can meet the needs of thousands of STUs online and interconnected at the same time. XMPP protocol has been standardized by Internet Engineering working group, and core protocols (such as RFC 6120, RFC 6121, and RFC 6122) have been released and updated. XMPP core specification has built-in relatively sound security mechanism. The IEC 61850 8-2 standard which is being developed by IEC TC57 organization adopts the XMPP mapping method to solve the network security issue.

As shown in Figure 2, XMPP supports mode applications of client/server (C/S) and server/server (S/S) and can also communicate with external networks through gateways.

TABLE 1: Technical requirements for distributed FA.

| Types of distributed FA | Information interaction time (ms) | Fault upstream switch isolation time (ms) | Recovery time of nonfault area (s) | Signal up time (s) |
|-------------------------|-----------------------------------|---|------------------------------------|--------------------|
| Quick distributed FA | ≤ 20 | ≤ 200 | ≤ 5 | ≤ 3 |
| Slow distributed FA | ≤ 200 | ≤ 5000 | ≤ 45 | ≤ 3 |

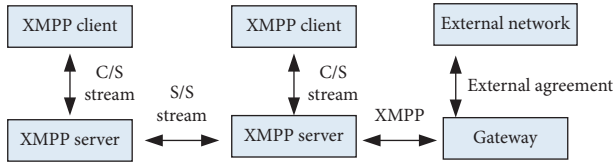


FIGURE 2: Typical network architecture of XMPP.

Communication between XMPP clients needs to be connected with XMPP server and forwarded by XMPP server:

- (1) The client establishes a connection with the server through TCP/IP and optionally sets the encryption option of transport layer security (TLS) to ensure the security of transport flow information
- (2) The client and server use simple authentication security layer (SASL) to obtain identity authentication
- (3) Open the XML stream and bind the client resources to form a complete identification JID (Jabber ID)
- (4) The client makes the JID of target address, and after the server looks up and authentication, a session between clients is establish. The specific message fragments are encapsulated in the middle of the stream in the form of XML stanza and transmitted in the form of XML stream. XMPP defines three different XML stanza-`<iq/>`, `<presence/>`, `<message/>`, to achieve different functions.

3.2. Distributed FA System Architecture Based on XMPP Mapping. The distributed FA system applies XMPP, as shown in Figure 3. STUs can be used as both the IEC 61850 client and the IEC 61850 server, but both of them are XMPP clients for XMPP communication. They need to be connected to the XMPP server set up in the communication network through TCP/IP protocol, and the server will transmit them to realize the conversation between clients. The configuration of the server can be selected according to the size of the system and the light and heavy load the server bears, for example:

- (1) Set up a single server in the master station or run XMPP server application in the front-end processor
- (2) Subregional settings, such as configure the server by feeder group

Cooperative mode distributed FA and agent mode distributed FA differ in the number of STUs participating in decision-making control, resulting in different data flow and data transmission volume. When a fault occurs on distribution lines, the data transmission capacity of collaborative mode distributed FA and agent mode distributed FA is similar in fault isolation and recovery. However, during fault

location, because collaborative distributed FA requires adjacent STUs for two-way interaction, there are many times of forwarding through the server and the server processes a large amount of information. The agent mode distributed FA only needs to be transmitted from the slave STU to the agent STU, and there is less interactive data forwarding. The real-time performance of the agent mode distributed FA is better than that of the collaborative distributed FA.

3.3. Service Mapping of XMPP. When XMPP is used for data transmission, the size of common data packets is usually several thousand bytes; since it has been transferred, encrypted, and decrypted through XMPP server, the actual transmission delay may be large in case of network blocking or large data packets; in this paper, XMPP is not used to transmit real-time control data with high real-time performance requirements; XMPP is used to transmit nonfault information model data, real-time measurement data, and historical data. The data are encapsulated in the XML stream in the form of XML stanza. After establishing the TCP/IP link, through the forwarding of the XMPP server, XML stream transmission from the STU to the STU is completed. The types of XML message format include `<iq/>`, `<message/>` and `<presence/>`.

When XMPP is used for information model data mapping, the most commonly used information exchange models are DataSet, Report, and Log. Among them, DataSet defines data values and data attribute values of logical nodes, and when the Report monitors that the information changes, that is to say, when the trigger condition is reached, the Report will immediately send the data set members to the client, and the whole process is recorded by Log; when the real-time measurement data and historical data are transmitted, the client completes the data transmission from the client to the client by forwarding the data set through the server in the XML stream. The association established before mapping uses two party application association of end-to-end information flow control.

Abstract communication service interface (ACSI) has no communication function and does not specify specific message format and encoding/decoding syntax [13]; therefore, IEC 61850 maps the information model and services of ACSI to specific communication service mapping (such as MMS and XMPP), in which MMS uses ASN.1 to make corresponding format regulations for service coding, and the coding format is BER, while XMPP mapping also uses similar data unit structure, but the coding mode is XER, and the specific mapping relationship is shown in Table 2.

3.4. Safety Protection of XMPP. Distributed FA uses XMPP for data transmission, mainly in the form of XML stream between the master station and the STUs or between the

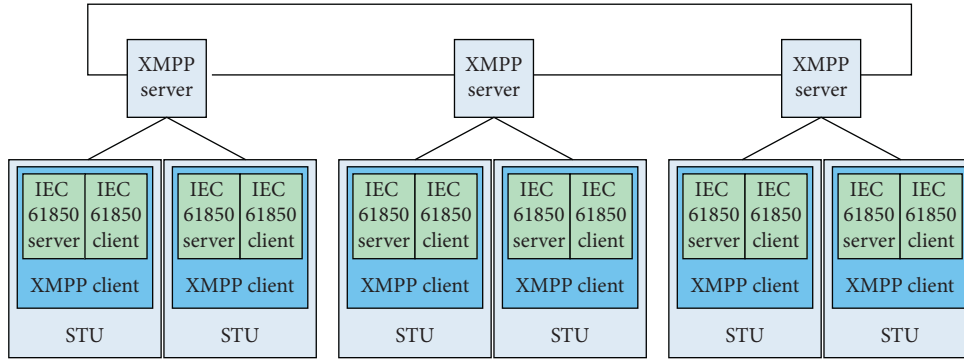


FIGURE 3: The architecture of XMPP in distributed FA systems.

TABLE 2: Distributed feeder automation related ACSI mapping table.

| IEC 61850 object | ACSIE service | ASN.1 BER of MMS | XML stanza and type of XMPP |
|------------------|--|---------------------------------------|-----------------------------|
| Associate | Associate | Initiate-requestPDU | IQ Type-set |
| | | Initiate-responsePDU | IQ Type-result |
| | | Initiate-errorPDU | IQ Type-result |
| Data | GetDataValues | Read-requestPDU | IQ Type-get |
| | | Read-responsePDU | IQ Type-result |
| | SetDataValues | Write-requestPDU | IQ Type-set |
| | | Write-responsePDU | IQ Type-result |
| | GetDataDefinition | GetVariableAccessAttribute-requestPDU | IQ Type-get |
| | GetVariableAccessAttribute-responsePDU | IQ Type-result | |
| Report | Report | InformationReport-requestPDU | Message Type-normal |
| | GetBRCBValues | Read-requestPDU | IQ Type-get |
| | | Read-responsePDU | IQ Type-result |
| | SetBRCBValues | Write-requestPDU | IQ Type-set |
| | Write-responsePDU | IQ Type-result | |
| Control | Operate | Write-requestPDU | IQ Type-set |
| | | Write-responsePDU | IQ Type-result |
| LCB | SetLCBValues | Write-requestPDU | IQ Type-set |
| | | Write-responsePDU | IQ Type-result |
| File | SetFile | ObtainFile-requestPDU | IQ type-get |
| | | ObtainFile-responsePDU | IQ Type-result |
| Data set | GetDataSetValues | Read-requestPDU | IQ Type-get |
| | | Read-responsePDU | IQ Type-result |
| | DataSetValues | Write-requestPDU | IQ Type-set |
| | | Write-responsePDU | IQ Type-result |

STU and the STU. During the transmission process, it may be subject to external malicious tampering, so this process needs effective security protection. XMPP itself contains two security mechanisms of transport layer security (TLS) and simple authentication security layer (SASL). Among them, TLS is used to encrypt the communication channel to ensure the information security of the data flow from the client to the server or from the server to the server; SASL is used to authorize the user, and multiple authentication mechanisms are included to ensure the safety of the transmission of information.

XMPP has two built-in security mechanisms; among them, TLS is divided into two layers. The three protocols of handshake, password specification change, and alarm contained in the upper layer, respectively, have the functions of identity authentication, security parameter negotiation and

change notification, flow closing, and error alarm, which can ensure the security of communication; the lower record layer protocol can encrypt and decrypt data, decompress and compress data, and check data integrity to ensure data security; it uses STARTTLS extension. The sender sends <starttls/>command to indicate the start of STARTTLS negotiation. The receiver uses <processed/> or <failure/> to reply.

Since authentication information needs to be sent during SASL negotiation, STARTTLS negotiation needs to be completed before SASL negotiation; SASL provides GSSAP, DIGEST-MD5, SCRAM (SCRAM-SHA-1 and SASL-SCRAM-SHA-1-PLUS), PLAIN and other mechanisms to realize authentication. In distributed FA, authentication between STU and STU or between STU and master station is realized by SASL built-in XMPP, and the security of transmitted data is protected by TLS lower layer, identity

authentication and error alarm are realized by TLS upper layer, and the security of data transmission is realized by XMPP built-in security protection.

4. Security Protection Scheme of GOOSE for Real-Time Data Transmission

In the distributed FA, this paper uses GOOSE to complete the transmission of fast real-time data. In order to realize the fast transmission of the message in the IP layer, the way of GOOSE over UDP is adopted, and it has the characteristics of based on peer-to-peer communication, fewer protocol control options, short message delay time, and fast transmission speed. At the same time, it meets the real-time performance requirements of distributed FA for control commands.

4.1. The Information Transmission of GOOSE over UDP. GOOSE over UDP adopts the publisher/subscriber mechanism. In order to ensure the data of real-time performance, priority is set in the Type of Service (TOS) field of the IP protocol in the network layer. TOS is considered to be composed of differentiated service code point (DSCP) and explicit congestion notification (ECN); in order to ensure the reliability of the data, the fast multiple retransmissions mechanism is adopted. At the same time, whether the message is lost or whether the communication is interrupted can be judged according to the allowable lifetime of the message. According to the status number (StNum) and the sequence number (SqNum), it can be judged whether the transmitted message has frame loss, wrong sequence, or repetition, and for more important information (such as switch action), double frame receiving mechanism is adopted to ensure the reliability of transmission information.

When using GOOSE transmission mechanism to realize FLISR function, such as agent mode distributed FA, multicast application association is adopted for communication, fault indication DataSet is sent to service restoration controller (SRC) through Report service for fault section judgment, and SRC completes fault isolation and recovery of nonfault section through Operate service [14].

4.2. The Security Protection of GOOSE over UDP. Distributed FA is used for fault handling, which has high requirements for the real-time performance, reliability, and safety of transmission messages. In order to ensure that the distributed FA can quickly and accurately implement the FLISR function, effective security protection is required for GOOSE messages.

In order to solve the security issue of GOOSE communication in substation, IEC 62351 recommends the authentication algorithm based on message authentication code (MAC). The MAC shall be generated through the computation of a 32 bit FCS calculated by ISO/IEC 13239 (ISO HDLC). Message digest is signed by RSASSA-PKCS1-V1_5 algorithm specified by RFC 2437 to generate digital signature with security encryption. In the work by Farooq's team [15], the encryption algorithm in IEC 62351 standard is tested for the encryption and decryption performance of GOOSE data.

The encryption and decryption time is 4.31 ms when the CPU is Intel i5-3210M, the main frequency is 2.5 GHz, and the GOOSE data packet is 256 bytes. At present, the CPU speed of distribution terminal is relatively low. Taking the STUs of Kehui's PZK-360H as an example, the main frequency of STUs is 454 MHz, and the encryption and decryption time is 24.303 ms after conversion according to the CPU speed. If the GOOSE data packet exceeds 256 bytes, the encryption and decryption time is longer. Therefore, the real-time performance does not meet the requirement that information interaction time is less than 20 ms in the fast distributed FA. In addition, the memory overhead of the encryption algorithm recommended by IEC 62351 is also large.

In order to solve the security issue of real-time control data in distributed FA, this paper uses the authentication of one-time signature based on HORS to enable GOOSE over UDP to detect whether the message is complete and whether it is intruded. The one-time signature is based on one-way function without trap gate, which has asymmetric secret information. At the same time, it has low requirements for hardware equipment, and it is fast in generating and verifying signature, which makes the one-time signature suitable for multicast authentication. GOOSE uses the one-time signature for the transmission of multicast data as shown in Figure 4.

SRC is used as key distribution center (KDC). KDC protocol based on RFC 3547 allows to support one-time signature algorithm. At the same time, in order to ensure that the key will not be stolen or tampered, KDC needs to update the key and refer to the handling method of the key in IEC 61850 90-5; the key update can be divided into two types: regular update and irregular update. The regular update is the update of the key under normal conditions, and the time is generally set as 30 min to 48 h. Because the computing power of STUs is relatively weak, the key length is relatively short. In order to ensure sufficient security, the maximum key lifetime is set as 30 min. The key generated by KDC is sent to each STU through UDP/IP multicast. After receiving the key, the STUs will save it. When using GOOSE for message transmission, it will be added to the message. The order of using the key is opposite to that of generating the key. The specific signature process is as follows [16–18]:

- (1) Key generation: generate t random n bit strings (s_1, s_2, \dots, s_t) , which form the private key S_K . The public key is then computed as $P_K = (v_1, v_2, \dots, v_t)$, where $v_i = f(s_i)$ and f is a one-way function.
- (2) Signing: to sign a message M , let $h = H(M)$, where H is a hash function. Split h into k substrings (h_1, h_2, \dots, h_k) of $\log_2 t$ bits each. Interpret each h_j as an integer i_j . The signature of message M can be expressed as $(s_{i_1}, s_{i_2}, \dots, s_{i_k})$, $1 \leq j \leq k$.
- (3) Verification: the recipient verifies the signature of the message M sent by the sender, uses the method in (2) to calculate and generates $(s'_{i_1}, s'_{i_2}, \dots, s'_{i_k})$, compares and verifies with the original signature, and check if $f(s'_j) = v_{i_j}$ holds.

Once the signature of message M is generated, it cannot be changed, but the tamper can tamper with the message by

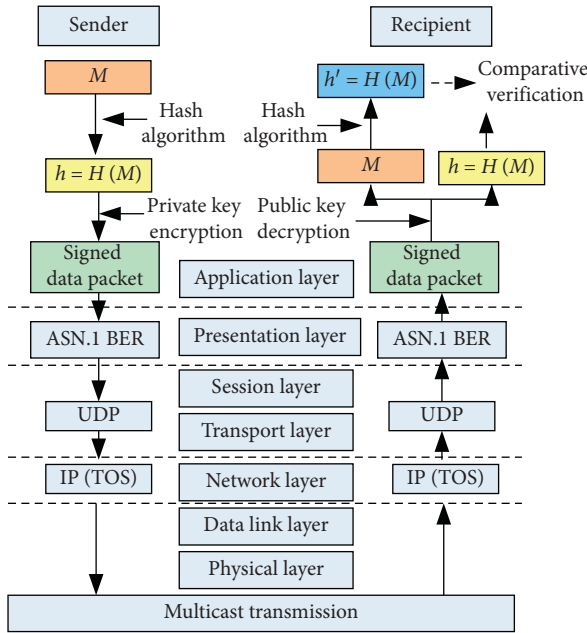


FIGURE 4: GOOSE multicast authentication data stream based on one-time signature.

forging the same hash value as message M . Figure 5 shows the case of tampering with information when k is taken as 3; the tamper changes message M to M' , then we need to match the correct hash value from the $K!$ Hash values, and so we can increase the security of one-time signature by increasing the size of K , but the calculation cost and signature size will also increase.

4.3. The Influence of Distributed FA Control Mode on Message Encryption. The control mode of distributed FA is divided into collaborative distributed FA and agent mode distributed FA. Because the STU of the cooperative mode distributed FA needs to communicate with the remaining $n - 1$ STUs, and in order to ensure the security of the keys of STU, the STUs communicating with each other must contain keys that can only be identified by each other and KDC needs to send $n \times (n - 1)$ keys. There is a large demand for the number of keys, a large amount of calculation and update work, and a high memory occupation of the STU; and the agent mode distributed FA takes the SRC as the KDC, SRC is responsible for managing and distributing the keys, and STU only needs to communicate with the SRC, so the number of keys is only n , and the number of keys is small. Considering the computing and storage capacity of the STU, agent mode distributed FA is more suitable for key distribution and management than collaborative distributed FA.

5. Experiment Test

The test is conducted for the agent mode distributed FA. The real-time control data between STUs and SRC are transmitted by GOOSE, and other data are transmitted by XMPP, as shown in Table 3.

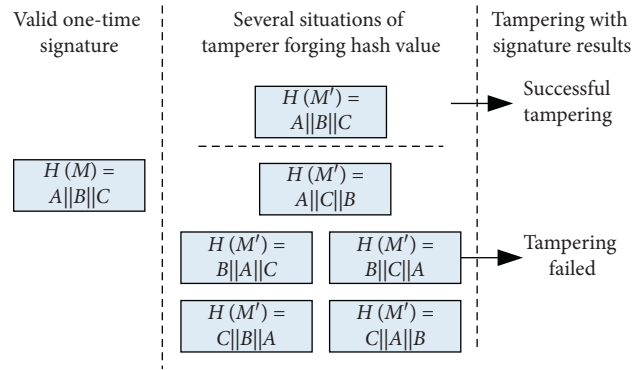


FIGURE 5: Tamperer forged signature case.

5.1. XMPP Message Test. In order to test the real-time transmission performance of IEC 61850 information by XMPP, the communication test system as shown in Figure 6 is built, which is composed of optical Ethernet switch, router, PC, and relevant application software. PC1 installs the XMPP universal server Openfire. PC2PC7 are XMPP clients. The STU1 and STU2 are used for the XMPP data test. The STUs use Kehui’s PZK-360H and built-in MPC 287 communication board with the main frequency of 454 MHz CPU, and the test program is developed in C++ under the embedded Linux operating system environment. PC4, PC5, PC6, and PC7 are used to simulate the other online users on the server in the distribution network, so as to realize the user login and the communication between the users, so as to generate the background traffic in the communication network. There are 502 simulated users in the test. The client program is implemented based on Java language, and the XMPP class library uses smack library.

The method of ping-pong test is adopted in communication delay test. First, record the time when STU1 sends a data packet as t_1 ; STU2 receives the data packet from STU1, then sends it back to STU1, and records the time when STU1 receives the returned data packet as t_2 , and then the end-to-end data transmission delay is obtained by calculating the time difference between t_1 and t_2 divided by 2. Each message of different sizes sends 5000 packets. The measured end-to-end data transmission delay includes network transmission delay and server processing delay. Packets sent to the server need to wait in the packet queue processed by the server and be forwarded to the destination address by the server. If the transmitted data package is encrypted, because the key of different STUs are different, it needs to be decrypted first and then encrypted and forwarded in the XMPP server. Therefore, the processing delay of the XMPP server mainly includes the encryption and decryption delay of the server and the forwarding delay of the server.

It can be seen from Figure 7(a) that the communication message size has a great influence on the transmission delay, which basically shows that the transmission delay increases with the increase of the number of bytes in the message. The average delay of encryption with security is higher than that of encryption without security, mainly because of the encryption and decryption delay of the server. Based on XMPP to transmit IEC 61850 data objects, it is necessary to combine

TABLE 3: Communication content and security protection of distributed feeder automation.

| Communication object | Transmission mode | Transmission content | Safety protection |
|----------------------|-------------------|---|--------------------|
| STU and SRC | GOOSE (fault) | Teleindication (signal of successful fault location, signal of successful switch disconnection, and signal of successful fault isolation) Remote control (switch control command, etc.) Telemetry (fault current value, etc.) | One-time signature |
| | XMPP (nonfault) | Configuration, telemetry, teleindication | TLS and SASL |
| SRC and MS | XMPP | Configuration, telemetry, teleindication | TLS and SASL |
| STU and MS | XMPP | Configuration, telemetry, teleindication | TLS and SASL |

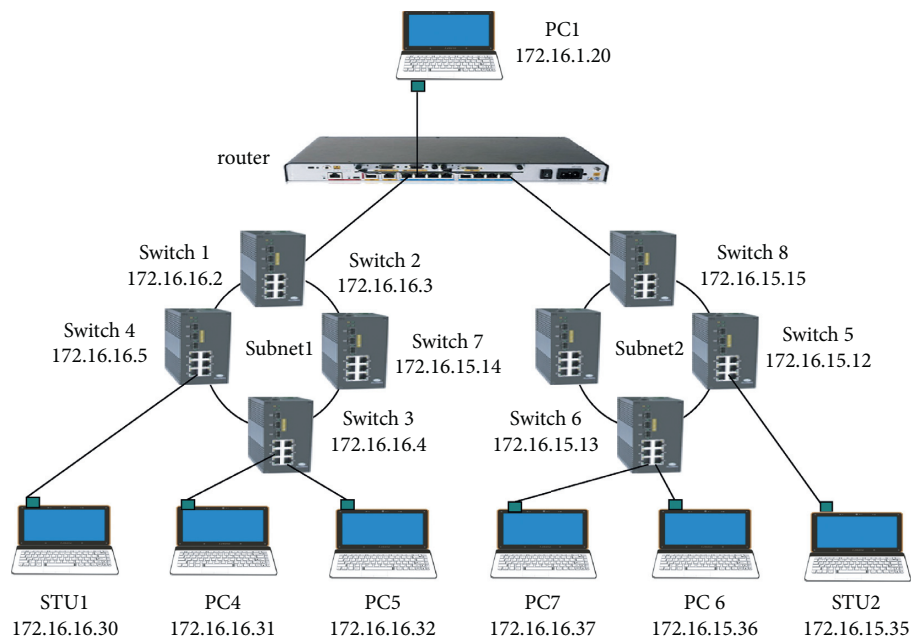


FIGURE 6: The test system of XMPP.

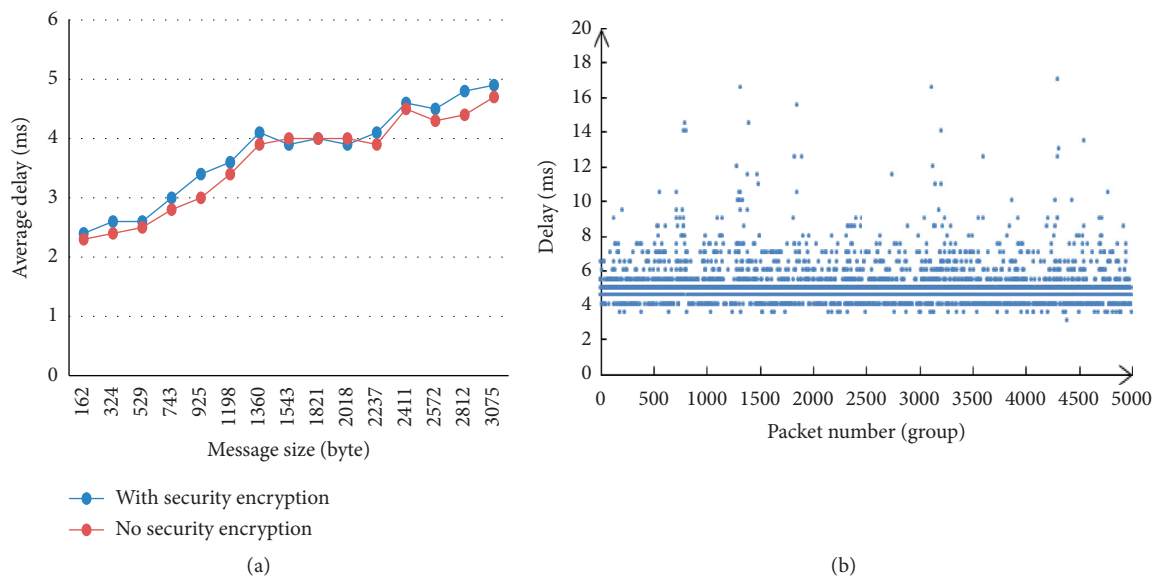


FIGURE 7: XMPP transmission delay test: (a) end-to-end transmission delay of packet size; (b) the scatter plot of the transmission delay test with a packet size of 3075 bytes.

the data types and service parameters of IEC 61850 to expand based on XML to form multilayer definition tags to encapsulate data. Therefore, a typical XMPP encapsulated data packet is generally more than a few kilobytes. Taking 3075 bytes message as an example, the average end-to-end transmission delay based on XMPP with security encryption is about 4.9049 ms, and the maximum delay is about 17 ms. It can be seen from Figure 7(b) that the transmission delay under this message is mainly concentrated in 4–6 ms, and the large point of delay value is discontinuous.

The encryption delay of the message is tested on the PC with the CPU main frequency of 2.2 GHz, and the measured time is 0.3 ms. Converted to the STU using Blackfin as the processor and the main frequency of the main board as 300 MHz, its encryption delay is about 2.25 ms. Therefore, the transmission delay based on XMPP transmission from encrypting the sending end of the STU to receiving the message at the receiving end of the STU is about 7.5149 ms.

In the work by Fan's team [10], it is introduced that the distributed fast message of the distribution network is transmitted by GOOSE mechanism, and it is measured that the distributed measurement and control message based on GOOSE over UDP scheme is within 1.5 KB, and its transmission delay is within 2 ms. In contrast, XMPP-based distributed measurement and control messages have a relatively large transmission delay when the message is 1500 bytes, and the average delay is about 3.9 ms. The main reason is that XMPP packets need to be forwarded by the server to ensure security. In addition, because XMPP adopts XML plain text encoding form, when transmitting the same IEC 61850 information content, XMPP message is larger, so the delay is correspondingly increased.

The China National Standard [12] stipulates that the following: (1) the delay time of fault information interaction of peer-to-peer communication for fast distributed FA shall not be greater than 20 ms; (2) the delay time of fault information interaction message of peer-to-peer communication for slow distributed FA shall not be more than 200 ms. According to the test, the maximum end-to-end transmission delay of 3075 bytes' message during security encryption is 17 ms. But in actual application on-site, the size of XMPP data packet may be larger than 3000 bytes. Compared with the actual communication environment, the test network environment is stable and the server performance is better. Therefore, the maximum transmission delay of XMPP message in actual transmission may exceed 20 ms, but it meets the requirements of slow distributed FA for the transmission delay of information exchange, so it can be applied in slow distributed FA based on XMPP. It cannot meet the demand of fast distributed FA.

5.2. The Real-Time Performance Test of GOOSE over UDP Message. In order to test the real-time performance of using GOOSE over UDP to transmit control messages, a distributed FA test system is built as in Figure 8, the test platform is composed of the active static simulation platform of distribution network, communication network, STUs and PCs. The active static simulation platform

of distribution network can simulate the failure of hand-held overhead line and cable line, and connect the voltage and current to the STUs through the corresponding transformer. The communication network consists of two SICOM3000 Ethernet switches, one router and single-mode optical fiber. The STUs uses five sets of PZK-360H from Kehui, and the PC is used to generate the background traffic in the network. In case of short-circuit fault of hand-held ring network, FLISR function is realized by GOOSE message communication between STUs. The time of fault isolation and recovery can be obtained by recording the trip contact signal output by the STUs and the fault simulation start contact signal time. Network message recording tool, Wireshark, is used to monitor the network and analysis GOOSE messages.

Since the STUs do not have the ability of high-precision time synchronization, it cannot directly test the transmission time of a single message between STUs. For the test of the transmission delay of GOOSE messages, the transmission delay can be realized by the ping-pong test (that is, the time difference between the sender's request message and the sender's acceptance of the response message divided by 2 as the transmission time) method after modifying the test program test.

The developed distributed FA test system uses the fault indication and control data set transmitted by GOOSE over UDP when a failure occurs; its message length is 335 bytes. Therefore, using the ping-pong method to test the GOOSE message transmission delay and encryption and decryption time are all for the 335 bytes GOOSE packet.

The failure test was carried out for the agent mode distributed FA, and the test was repeated 10 times, the signal of the switch node is recorded by the wave recorder to obtain the fault isolation time, and the log recorded in the STUs is used to obtain the encryption and decryption time, the fault detection time, and the switch control time. After the test, the total isolation time in agent mode distributed FA is 172.766 ms on average; among them, the average time from the over-current start to the protection information is written into the STU log as 17 ms. The average time from STUs receiving a remote control command to sending a remote control command to the switch is 19 ms, and the average time from STUs sending a remote control command to STUs detecting that the switch is off is 86 ms.

The encryption and decryption time of GOOSE message is shown in Table 4, and the average value of encryption and decryption time is 6.344 ms. The research group conducted a GOOSE over UDP transmission delay test with the message length of 335 bytes [10], and the average value of the transmission delay was 0.539 ms, so the transmission delay of the GOOSE messages with security encryption is about 6.883 ms. In Table 4, the transmission delay of STUs encryption is relatively smaller than that of decryption, because the separated message M needs to be calculated using the hash algorithm when decrypting; and then the obtained digest h' is compared with the original digest h to verify the correctness of the signature. Encryption does not contain the separation and verification process, so the decryption time is slightly longer than the encryption time.

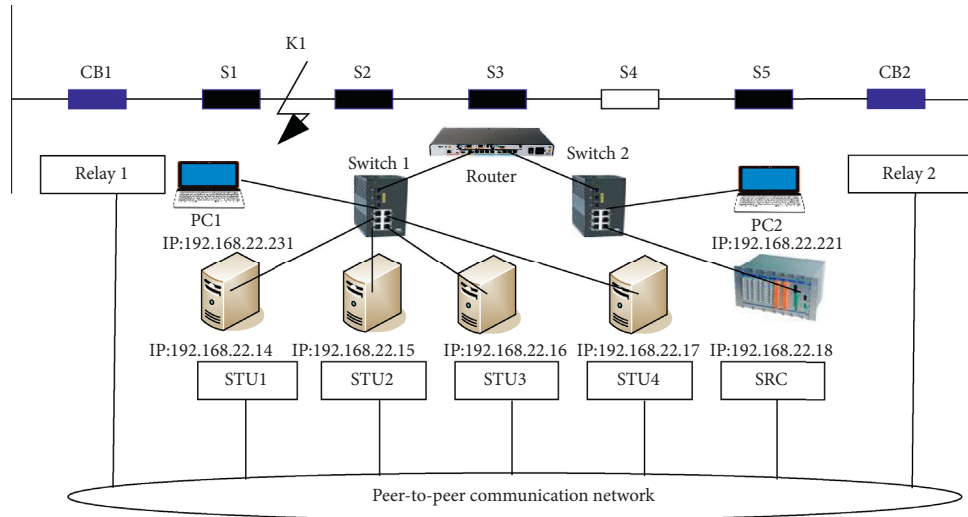


FIGURE 8: Distributed FA test system.

TABLE 4: Encryption and decryption time of GOOSE messages of STU.

| Encryption and decryption | Max. (ms) | Min. (ms) | Ave. (ms) |
|---------------------------|-----------|-----------|-----------|
| Encryption | 6 | 2 | 2.645 |
| Decryption | 7.125 | 2.809 | 3.699 |

For the transmission delay of GOOSE messages, it is not encrypted when the ping-pong method is used for testing, so as to reduce the processing delay of the STUs as much as possible. The test results are basically consistent with those of the work by Fan's team [10], which will not be discussed here.

PC2 is used to send data packets to PC1 to generate constant network background traffic, and different network load rates are generated by controlling the transmission rate. As shown in Figure 9, when the network load rate is less than 97%, the average transmission delay of the message is less than 7.3 ms and is less affected by the network background traffic, and when the network load rate is greater than 97%, the transmission delay increases sharply. In the distributed FA communication system of the distribution network, the network load rate generally does not exceed 30% [10], so the network load rate has little impact on the communication of the distributed FA.

In the process of distributed FA fault handling, real-time data are transmitted by GOOSE over UDP, and the Report after fault is transmitted to the master station through XMPP.

Based on the scheme developed in this paper, the average total isolation time of the test is 172.7 ms, which can basically meet the requirements of the fast distributed FA. In the total isolation time of distributed FA, the encryption and decryption time of GOOSE message and the communication delay of data packet account for a relatively small proportion. The main delay factors are fault detection and control strategy. Due to the need to transmit data to SRC for decision-making and then sending control commands to STUs for execution, the fault

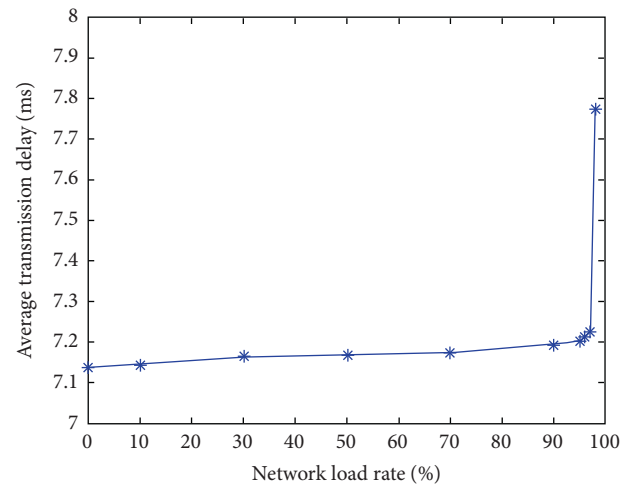


FIGURE 9: Transmission delay of different network load rates of GOOSE with secure encryption.

processing speed of agent mode distributed FA is slower than that of collaborative distributed FA, but the key distribution processing of agent mode distributed FA is easier than that of collaborative distributed FA.

6. Summary

In order to standardize the communication mapping of distributed FA and solve the issue of communication security protection between STUs, this paper studies the IEC 61850 communication mapping issue of XMPP in the application of distributed FA. XMPP based on XML extension and built-in security mechanism can realize data encryption and integrity transmission. However, XMPP packets are transferred and encrypted in the server to a certain extent, which affects the real-time performance of transmission, and its real-time performance cannot meet the requirements of fast distributed FA. Realization of fast control data

transmission through GOOSE over UDP can meet the real-time requirements of fast distributed FA. Due to the relatively weak computing capabilities of the STUs, the GOOSE message is protected by a one-time signature algorithm based on HORS with a small amount of calculation.

The test system is developed based on this scheme. The test results show that the transmission delay of XMPP can meet the requirements that the information interaction delay of slow distributed FA is no more than 200 ms, and the user authorization, authentication, and communication channel encryption technology based on SASL and TLS can realize the horizontal and vertical information security protection of distribution network, which provides a safe and effective communication mapping method for distributed control applications such as STUs and access of DERs. The real-time performance of GOOSE over UDP with increased security control can basically meet the requirements that the information interaction delay of fast distributed FA.

This paper conducts a preliminary study on the communication security of distributed FA; considering distribution and management for the key, the scheme is more suitable for the application of agent mode distributed FA. So, the agent mode distributed FA control is implemented at present, and the real-time performance is better when the collaborative distributed FA is adopted, but the management and distribution of the key are complicated. The next stage will further study the security control of the collaborative distributed FA.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work was supported by the Key Project of Smart Grid Technology and Equipment of National Key Research and Development Plan of China (No. 2016YFB0900600) and Technology Projects of State Grid Corporation of China (No. 52094017000W).

References

- [1] Y. Chen, Z. Zhu, B. Xu et al., "The use of IEC61850 for distribution automation," in *Proceedings of the 2016 China International Conference on Electricity Distribution (CICED 2016)*, pp. 10–13, Xian, China, August 2016.
- [2] T. Yip, J. Wang, B. Xu, K. Fan, and T. Li, "Fast self-healing control of faults in MV networks using distributed intelligence," *CIREN-Open Access Proceedings Journal*, vol. 2017, no. 1, pp. 1131–1133, 2017.
- [3] W. Ling, D. Liu, Y. Lu et al., "Model of intelligent distribution feeder automation based on IEC 61850," *Automation of Electric Power Systems*, vol. 36, no. 6, p. 9095, 2012.
- [4] Y. Chen, S. Dai, C. Yang et al., "IEC 61850-based modeling of intelligent distributed feeder automation system," *Electric Power Automation Equipment*, vol. 36, no. 6, pp. 189–222, 2016.
- [5] S. M. S. Hussain, M. A. Aftab, and I. Ali, "IEC 61850 modeling of DSTATCOM and XMPP communication for reactive power management in microgrids," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3215–3225, 2018.
- [6] H. Wang, D. Xiong, P. Wang, and Y. Liu, "A lightweight XMPP publish/subscribe scheme for resource-constrained IoT devices," *IEEE Access*, vol. 5, pp. 16393–16405, 2017.
- [7] C. S. Cho, W. Chen, C. Liao et al., "Building on the distributed energy resources IoT based IEC 61850 XMPP for TPC," in *Proceedings of the 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*, IEEE, Taiwan, China, May 2019.
- [8] X. Hou, Y. Chen, B. Xu et al., "Application of extensible message and presence protocol in distributed feeder automation," *Automation of Electric Power Systems*, vol. 43, no. 4, pp. 228–236, 2019.
- [9] X. Chen, B. Xu, Y. Chen et al., "Real-time data fast transmission technology for distributed control of distribution network," *Power System Protection and Control*, vol. 44, no. 17, pp. 151–158, 2016.
- [10] Y. Fan, Q. Wang, H. Peng et al., "GOOSE over UDP transmission mechanism for real-time data fast transmission in distribution network," in *Proceedings of the Green & Sustainable Computing Conference*, Orlando, FL, USA, October 2017.
- [11] C. Tang, Z. Yang, B. Song et al., "A method of intelligent distributed feeder automation for active distribution network," *Automation of Electric Power Systems*, vol. 39, no. 9, pp. 101–106, 2015.
- [12] GB/T 35732-2017, *Technical Specifications of Intelligent Remote Terminal Unit of Distribution Automation*, China Electric Power Press, Beijing, China, 2017.
- [13] L. He, *Getting Started with IEC 61850 Applications*, China Electric Power Press, Beijing, China, 2012.
- [14] Communication networks and systems for power utility automation: part 7-2 basic information and communication Structure-abstract communication service interface(ACSI): IEC 61850-7-2.2014.
- [15] S. M. Farooq, S. M. S. Hussain, T. S. Ustun et al., "Performance evaluation and analysis of IEC 62351-6 probabilistic signature scheme for securing GOOSE messages," *IEEE Access*, vol. 7, pp. 32343–32351, 2019.
- [16] Q. Wang, H. Khurana, Y. Huang, and K. Nahrstedt, "Time valid one-time signature for time-critical multicast data authentication," in *Proceedings of the IEEE Infocom*, pp. 1233–1241, Rio de Janeiro, Brazil, April 2009.
- [17] Q. Li and G. Cao, "Multicast authentication in the smart grid with one-time signature," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 686–696, 2011.
- [18] C. Ji, J. Kim, J. Y. Lee et al., "Review of one-time signatures for multicast authentication in smart grid," in *Proceedings of the International Conference & Expo on Emerging Technologies for A Smarter World*, Melville, NY, USA, October 2015.
- [19] T. Yip, J. Wang, B. Xu, K. Fan, and T. Li, "Fast self-healing control of faults in MV networks using distributed intelligence," *CIREN-Open Access Proceedings Journal*, vol. 2017, no. 1, pp. 1131–1133, 2017.
- [20] R. Kuntschke, M. Winter, C. Glomb et al., "Message-oriented machine-to-machine communication in smart grids," *Computer Science-Research and Development*, vol. 32, no. 1-2, pp. 131–145, 2017.

Research Article

Adaptive Particle Swarm Optimization with Gaussian Perturbation and Mutation

Binbin Chen ¹, Rui Zhang ², Long Chen ² and Shengjie Long ³

¹Graduate School, Xi'an International Studies University, Xi'an 710128, China

²School of Information Engineering, Zunyi Normal College, Zunyi 563002, China

³School of Traffic & Transportation Engineering, Central South University, Changsha 410004, China

Correspondence should be addressed to Shengjie Long; longshengjie12@csu.edu.cn

Received 10 October 2020; Revised 18 January 2021; Accepted 23 January 2021; Published 4 February 2021

Academic Editor: Wei Li

Copyright © 2021 Binbin Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The particle swarm optimization (PSO) is a wide used optimization algorithm, which yet suffers from trapping in local optimum and the premature convergence. Many studies have proposed the improvements to address the drawbacks above. Most of them have implemented a single strategy for one problem or a fixed neighborhood structure during the whole search process. To further improve the PSO performance, we introduced a simple but effective method, named adaptive particle swarm optimization with Gaussian perturbation and mutation (AGMPSO), consisting of three strategies. Gaussian perturbation and mutation are incorporated to promote the exploration and exploitation capability, while the adaptive strategy is introduced to ensure dynamic implement of the former two strategies, which guarantee the balance of the searching ability and accuracy. Comparison experiments of proposed AGMPSO and existing PSO variants in solving 29 benchmark functions of CEC 2017 test suites suggest that, despite the simplicity in architecture, the proposed AGMPSO obtains a high convergence accuracy and significant robustness which are proven by conducted Wilcoxon's rank sum test.

1. Introduction

Particle swarm optimization (PSO) is an evolutionary computing technique proposed by Kennedy and Eberhart in 1995 [1], originating from the simulation of predation and other behaviors of bird flocks and fish schools. The solution of each optimization problem in the algorithm is similar to a "particle" in the search space. The particle swarm algorithm randomly generates an initial swarm and gives each particle a random velocity. During the optimization process, the particles adjust the velocity and trajectory according to the experience of themselves and companions, so that the whole swam contains the ability to fly to a better search area. Involving few parameters and with easy implementation, PSO has been widely used in many fields such as function optimization, neural network training, fuzzy system control, pattern recognition, and engineering application. However, the PSO algorithm still has problems such as premature and easily falling into local optimum when tackling complex

multimodal problems. In order to improve the solving ability of particle swarm optimization, researchers have proposed methods, such as an adjustment of the inertial parameters of particle swarm algorithm, including dynamic policies and adaptive methods, learning factors, and social factors [2], a neighborhood searching strategy to strengthen the exploration of the neighborhood of the current population [3], an adoption of the information-sharing mechanism to enhance population diversity and avoid premature algorithm convergence [4], and the integrations with other algorithms, such as the combination of particle swarm optimization algorithm and immune algorithm, genetic algorithm, and artificial bee colony algorithm [5].

A variety of improved methods are proposed to solve the existing problems of PSO. The inertia weight and velocity parameters were dynamically adjusted through the particle swarm's convergence state to speed up the convergence speed and balance the global search and local search capabilities [6]. Alatas et al. [7] proposed a strategy to learn

from outstanding individuals other than the optimal particles to adapt to the solution of high-dimensional problems. Zhao et al. [8] introduced biology principles to give the particles the ability of multi-crossover and swarm colonization behaviors. Liang et al. [9] implemented a neighborhood development strategy to improve the algorithm's search ability. Chen et al. [10] leveraged the optimal information for all other particles to update the velocity of the particles in different dimensions. A particle swarm algorithm with lifecycle and challenging behavior is proposed to preserve particles' activity during the evolution of particle swarm, which is beneficial to the global range search [11]. Frans and Engelbrecht [12] incorporated chaos into the particle movement process, so that the particle swarm alternately moves between chaos and stability, gradually approaching the best point. Tian [13] initialized the particle swarm in a chaotic manner to ensure that the particle swarm can be evenly distributed in the solution space and achieve better global search capabilities. Du et al. [14] and Munlin and Anantathanavit [15] proposed to use multiple methods to realize the particles' evolution to improve the search capability of the algorithm. Kiran [16] improved the efficiency of the algorithm evolution with a new evolutionary mechanism. It improves the calculating speed of the algorithm and particularly has an advantage in solving multimodal problems.

These methods improve the performance of PSO to certain extent, but there are still many flaws, such as high complexity in architecture and low convergence speed. The solutions, such as adaptive, perturbation, and mutation, were incorporated to address these problems.

Aiming to attain the prominent performance of PSO, the adaptive strategy was introduced to dynamically update the parameters of the algorithm in prior studies, i.e., inertia weight [17–21], velocity and position [22, 23], and ω , c_1 , and c_2 of each particle [5, 24]. Wang et al. [25] and Li and Cheng et al. [26] introduced a mixed adaptive strategy to adjust the parameters in order to balance the search and convergence capabilities. Beyond adaptive updates the parameters, more complex adaptive strategies are proposed. The particles are randomized based on the detection of the changes of $gbest$ value [27]. In order to adaptively maintain the social attribution of swarm, the inactive particles are taken off based on the diversity of fitness between current particle and the best historical experience [28].

In order to promote particles to jump out of local optimum further improving the global searching ability, multiple perturbation strategies were introduced. A chaotic perturbation was incorporated into the PSO algorithm, which improved particles' diversity [29]. Mahmoodabadi et al. [30] utilized Cauchy perturbation and reverse learning to accelerate the particle swarm's convergence and escape the local optimal solution. Wang et al. [31] proposed nonuniform mutation and multistage perturbation of particles, which perturbs the optimal solution at different stages of evolution, thereby increasing group diversity and increasing the probability of jumping out of local extreme points.

For increasing vitality and diversity of particles, mutation strategy has been implemented in many optimizations of PSO. Pehlivanoglu [32] applied a mutation strategy into global random diversity and local controlled diversity. The undesired particles are replaced following the mutation strategy to accelerate the convergence speed [33]. Large-scale mutation and small-scale mutation are conducted to prevent the premature convergence while guaranteeing the convergence speed [34].

By contrast with the prior studies, we proposed an adaptive principle according to which the perturbation and mutation are conducted to balance the convergence accuracy and rate. Our main contributions are summarized as follows:

- (1) An adaptive adjusting rule is incorporated following the cosine law, in order that the particles are interfered with larger amplitude to improve the particle's global search ability in the early stage and with a smaller amplitude to improve the convergence accuracy.
- (2) Following the adaptive strategy, the Gaussian perturbation is incorporated to pump the optimal particle to jump out of the local optimum.
- (3) Identically, according to the adaptive strategy, the mutation is implemented to improve the diversity of particles that have stagnated evolution and to balance the ratio of inheritance and mutation to ensure the population's searching ability.

2. Adaptive Mutation Particle Swarm Optimization Algorithm with Gaussian Perturbation

2.1. Basic Particle Swarm Optimization. PSO first initializes a swarm. A particle in the swarm represents a solution of each search space, and each particle has two parameters: position and velocity. Assuming that the size of the current swarm $P(t)$ is N , the position, and the velocity of the i -th particle in the swarm are expressed as $v_i(t)\{v_{i1}(t), \dots, v_{iD}(t)\}$ and $X_i(t)\{X_{i1}(t), \dots, X_{iD}(t)\}$, where D is the dimension of the problem and t is an evolutionary algebra. The particles are evaluated by a previously designed fitness function. The particle i updates its velocity and position through the swarm optimal $gbest$ and the individual historical optimal $pbest_i$ in the iterative process. The equations for updating the velocity and position of a particle are as follows:

$$\begin{aligned} v_{id}(t+1) &= w \cdot v_{id}(t) + c_1 \cdot r_1 \cdot (pbset_{id} - x_{id}(t)) \\ &\quad + c_2 \cdot r_2 \cdot (gbset - x_{id}(t)), \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1), \end{aligned} \quad (1)$$

where w is the particle's inertia weight, which determines the degree of influence of the particle's previous velocity on the current velocity, c_1 is the particle self-cognition learning coefficient, c_2 is the social cognitive learning coefficient, and r_1 and r_2 are random numbers between 0 and 1.

2.2. Task Definition. As the number of iterations increases in the standard PSO algorithm, the particles will gradually approach the optimal solution, the evolutionary rate and swarm diversity will gradually decrease. Once the optimal particle falls into the local optimum, it hardly escapes. To address this problem, we incorporate the Gaussian perturbation and mutation strategy where the threshold $stop_num$ is set to define whether the particles are in the evolutionary stagnation state or not. Since fitness of a particle ceased to evolve, record the continuous times as tag, namely, the times of i -th particle as $tag(i)$ and the times of $gbest$ as $tag(g)$. If $tag(g) \geq stop_num$, it means that the evolution of the population has stagnated and the Gaussian perturbation is applied to pump the population to jump out of the local optimum. If $tag(i) \geq stop_num$, it means that the evolution of this particle has stagnated, and the mutation is conducted to update the particle. Either perturbation or mutation is conducted following the proposed adaptive strategy, guaranteeing the balance of searching ability and accuracy. The process of AGMPSO is shown in Figure 1.

2.2.1. Adaptive Strategy. In many previous studies, the amplitude of interference in PSO parameters remains the same during the iterations, which is not beneficial for the convergence in the later period. Aiming to reach the ideal state of PSO, a larger amplitude of interference is required at the early iterative stage to ensure a better global searching ability, and a smaller one in the late iterative stage to guarantee the convergence. Hence, a dynamically adaptive strategy is necessary. In this study, we introduce the probability Pc adaptively altering following the cosine law as iteration increases, and the equation is as follows:

$$Pc(t) = c_3 \cdot \left(1 + \cos\left(\frac{t \cdot \pi}{Max_Gen}\right) \right), \quad (2)$$

where t is current evolutionary iteration, $Pc(t)$ (as shown in Figure 2) is the probability of application of Gaussian perturbation or mutation of t iteration, c_3 is the adaptive coefficient, and Max_Gen is the maximum number of iterations.

2.2.2. Gaussian Perturbation Strategy. Gaussian perturbation is adopted in the particle $gbest$, which is in the evolutionary stagnation state, to improve the ability to jump out of the local optimum. In order to ensure that each dimension of $gbest$ has a possibility of escaping the local optimum, each dimension $gbest_d$ of $gbest$ is updated by Gaussian perturbation with probability $Pc(t)$, which guides the perturbation to conduct adaptively so as to ensure the better ability of escaping from the local optimum at early stages and also the better convergence ability during the later stages.

In addition to adapting the algorithm stages, via adaptive variance δ , the Gaussian perturbation strategy is capable of adapting the proposed PSO algorithm to different functions according to whose value spaces (as shown in equation (4)).

When the evolution of particle $gbest$ is stagnant, that is, $tag(g) \geq stop_num$, the perturbation is implemented as follows:

$$gbest_d = gbest_d \cdot r_3 \cdot guass_d, \quad (3)$$

$$guass_d = \text{Guassion}(0, \delta_d^2), \quad (4)$$

$$\delta_d = \frac{1}{5} \cdot (X_{\max} - X_{\min}), \quad (5)$$

where r_3 is a random number between 0 and 1, $gbest_d$ is the optimal particle of the d -th dimension, and δ_d is the adaptive variance of the d -th dimension.

2.2.3. Mutation Strategy. The mutation strategy is utilized to improve the particle diversity of the algorithm and balance the ratio of the mutation and the inheritance to ensure the convergence. Similar to the Gaussian perturbation strategy, the mutation strategy can adapt our algorithm to different functions as follows:

$$v_{id}(t+1) = w \cdot v_{id}(t) + c_1 \cdot r_1 \cdot (pbset_{id} - x_{id}(t)) + c_4 \cdot r_4 \cdot mut_d, \quad (6)$$

$$mut_d = X_{\min} + rand(X_{\max} - X_{\min}), \quad (7)$$

where c_4 is an adaptive mutation coefficient, r_4 is a random number between 0 and 1, and mut_d is the degree of adaptive mutation of d -th dimension.

When a particle falls into the evolutionary stagnation, the mutation operator is introduced into partial dimensions in the speed updated equation (6), which increases the diversity of the population getting rid of the constraints of $gbest$ particles, especially improves the search ability of particles with low speed due to converging near $gbest$, and promotes the particle utilization. Since the dimension d of the particle also uses the probability Pc to mutate, the mutated range of the algorithm's particles is large in the initial period, which is favorable to the global search. In the later period, the mutated range and the inheritance ratio are small, which is beneficial for algorithm convergence.

2.2.4. Algorithm Complexity Analysis. The computational costs of the standard PSO include the initialization $O(mn)$, fitness evaluation $O(mn)$, and velocity and position update $O(2mn)$ (m and n are the swarm size and dimension, respectively). Thus, the time complexity of the PSO is $O(mn)$. Compared with the standard PSO, AMGPSO involves two operators. However, the Gaussian perturbation operator $O(n)$ or the mutation operator $O(mn)$ need to be conducted separately only when the global best position is stagnant, or the personal best position is stagnant within several iterations. The worst-case time complexity of AGMPSO is $O(mn)$, including the initialization $O(mn)$, evaluation $O(mn+n)$, and update $O(2mn+mn)$. The conclusion can be drawn from the above component complexity analyses;

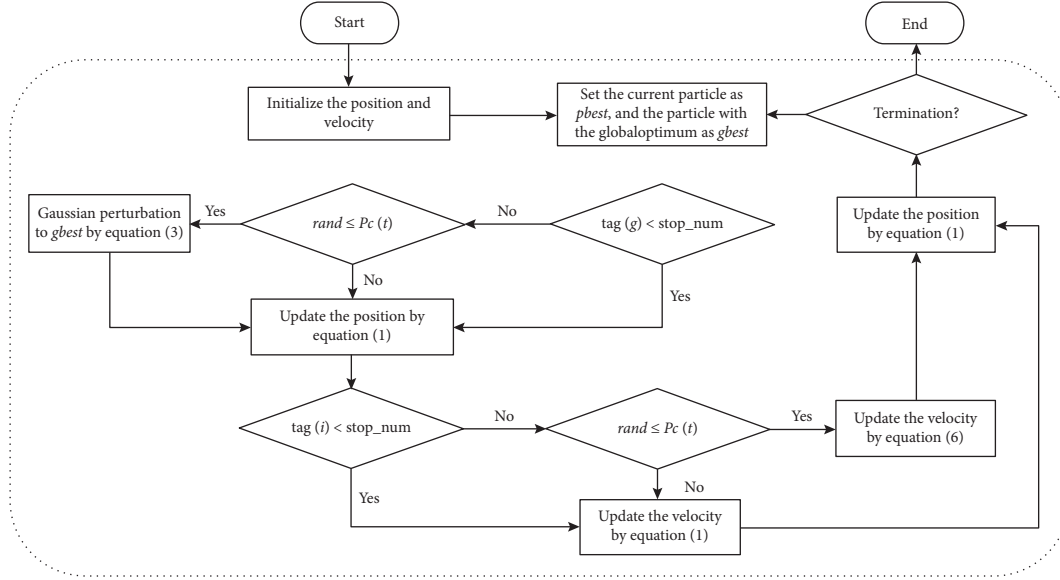
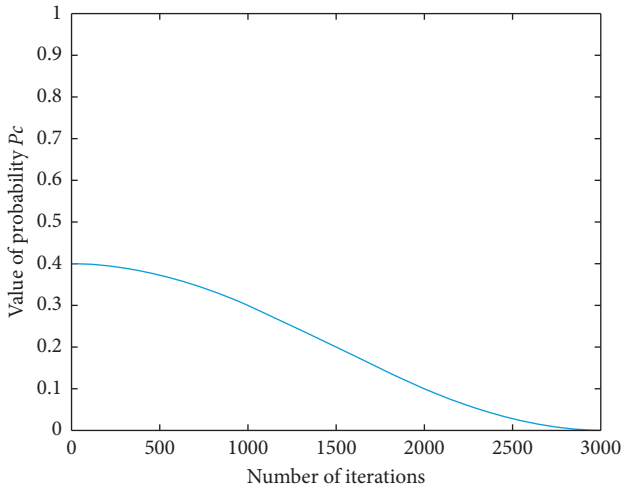


FIGURE 1: Flowchart of AGMPSO.

FIGURE 2: The curve of probability P_c during the whole iterations.

AGMPSO contains the same level of time complexity as the standard PSO algorithm.

3. Experiments and Discussions

3.1. Algorithm Aggregation Degree Analysis. The ideal status of PSO is that, in the early stage, the particles can explore the solution space more dispersedly, but in the later stage, they can better aggregate to obtain higher convergence accuracy. We introduce an aggregation degree to analyze the convergence status of the standard PSO and the AGMPSO. When the particle fitness deviation from the group average value is larger, and the particle aggregation degree θ is larger, the particle diversity is better, and the algorithm search ability is stronger. Aggregation degree θ of the t -th generation particle is expressed as follows:

$$\theta = \frac{1}{N} \cdot \sum_{i=1}^N \left| \frac{f(x_i^t) - f_{\text{avg}}^t}{f_{\text{max}}^t - f_{\text{min}}^t} \right|, \quad (8)$$

where f_{avg}^t is the average fitness value of t -th generation particle, f_{max}^t is the maximum fitness value of t -th generation particle, and f_{min}^t is the minimum fitness value of t -th generation particle.

Figure 3 shows the comparison of aggregation degree between the standard PSO and AGMPSO while solving the Rastrigin function. (A) and (B) are the aggregation curves of standard PSO and AGMPSO in 3500 iterations, and (C) and (D) are enlarged screenshots of the last 20 iterations. It can be seen that the standard PSO holds the higher particle aggregation even in later iterations, which indicates the high diversity, that is, the poor convergence. It is worth noting that the aggregation degree remains a high level, which means the swarm did not converge to a satisfactory extent till the end. In contrary, AGMPSO can maintain higher diversity throughout the early period and lower diversity in the later period, which ensures both the global search ability and the convergence.

3.2. Comparison with PSO Variants. In order to evaluate the performance of the proposed algorithm, the comparison experiments of AGMPSO with PSO, TSLPSO, HFPSO, and MPEPSO are conducted in this section, and parameters of each algorithm are listed in Table 1. All experiments were performed under Windows 10 system, eight-core processor (Intel (R) Core (TM) i7-10700K CPU @ 3.80 GHz), 16G memory using MATLAB R2018a.

3.2.1. Benchmark Functions. CEC 2017 [38] test suites are introduced in experiments, including the 29 benchmark functions divided into four categories: unimodal functions, simple multimodal functions, hybrid functions, and

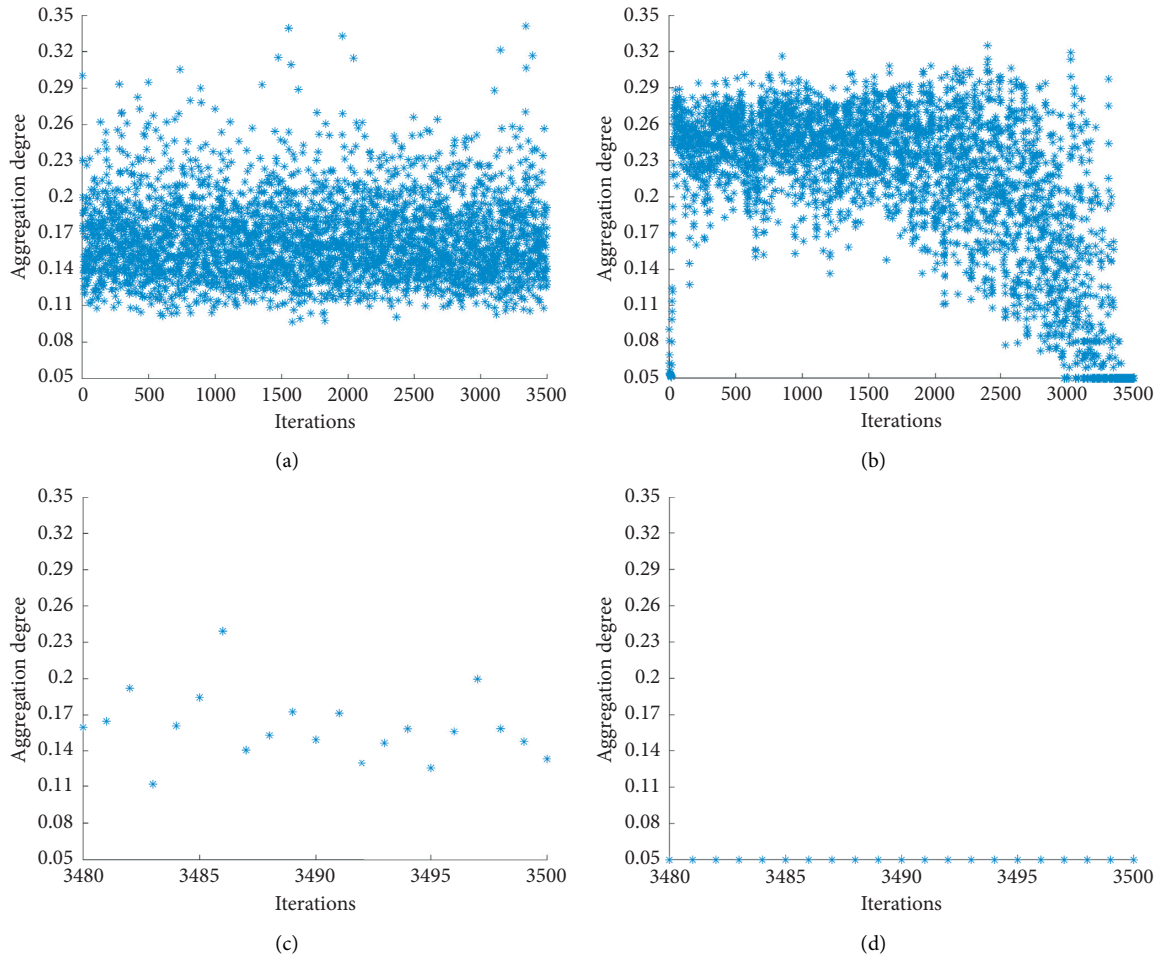


FIGURE 3: Aggregation degree of standard PSO and AGMPSO in solving Rastrigin function.

TABLE 1: The parameters of all comparison competitors.

| Algorithm | Parameter settings |
|-------------|--|
| AGMPSO | $w = 0.729, c_1 = c_2 = c_4 = 1.49445, c_3 = 0.2$ |
| TSLPSO [35] | $w = 0.9 \sim 0.4, c_1 = c_2 = 1.5, c_3 = 0.5 \sim 2.5$ |
| HFPSO [36] | $w_i = 0.9, w_f = 0.5, c_1 = c_2 = 1.49445, \alpha = 0.2, B_0 = 2, \gamma = 1$ |
| MPEPSO [37] | $w = 0.9 \sim 0.4, c_1 = c_2 = 0.5 \sim 2.5, LP = 10, \lambda_h = 0.1$ |
| PSO [1] | $w = 0.9 \sim 0.4, c_1 = c_2 = 2.0$ |

composition functions. Based on the description of the definitions of CEC 2017 test suits, F2 has been excluded because of its unstable behavior especially for higher dimensions.

Two series of experiments are performed, the dimension of each test function of each series is 10 and 30, respectively, the population size is 30, and each algorithm runs each benchmark function for 30 runs independently. According to the definitions of CEC 2017 test suits, the stop condition of each run is that the maximum number of function evaluations (MaxFES) reaches $10000 * D$, that is, $MaxFES = 100,000$ for 10D, $MaxFES = 300,000$ for 30D. Table 2 shows the search range and the global optimum of benchmark functions.

3.2.2. *Comparison of Simulation Results for Benchmark Functions.* The comparison results of mean values (mean), standard deviation (std), and Wilcoxon rank sum test (h) produced by all compared PSO variants on 10-dimensional tested functions are represented in Table 3 and 30-dimensional in Table 4, where the optimal results are marked in bold. The comparison results of computation time of each run are in Figures 4 and 5.

As shown in Table 3, AGMPSO performs well for unimodal functions, especially getting optimal mean fitness value on F3, despite the same result of TSLPSO on F3. In solving the multimodal functions from F4 to F10, AGMPSO has superior advantage, achieving better results on 5 out of 7 functions. However, AGMPSO does not obtain advantage solving the hybrid functions (F11–F20), and HFPSO gets the same achievement as proposed algorithm: 4 out of 10. It is still worth noting that the standard deviations of AGMPSO are smaller than HFPSO, which indicates more stability. For composition functions (F21–F30), AGMPSO algorithm perfectly reflects the better ability to search global optimum on 6 test functions, compared with the other competitors. In general, AGMPSO is top ranked on 14 out of all functions. Although on F3, F28, and F29, the proposed algorithm does not contain the statistically significant advantage, its mean

TABLE 2: Description of benchmark functions of CEC2017 test suits.

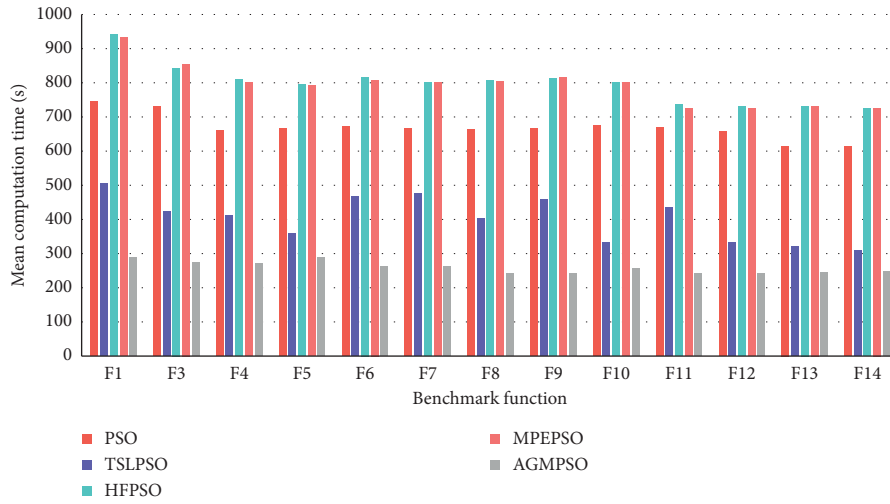
| No. | Function name | Search range | Global optimum | Type |
|-----|--|--------------|----------------|-------------|
| F1 | Shifted and rotated bent cigar function | (-100, 100) | 100 | Unimodal |
| F3 | Shifted and rotated Zakharov function | (-100, 100) | 300 | Unimodal |
| F4 | Shifted and rotated Rosenbrock's function | (-100, 100) | 400 | Multimodal |
| F5 | Shifted and rotated Rastrigin's function | (-100, 100) | 500 | Multimodal |
| F6 | Shifted and rotated expanded Scaffer's F6 function | (-100, 100) | 600 | Multimodal |
| F7 | Shifted and rotated Lunacek Bi_Rastrigin function | (-100, 100) | 700 | Multimodal |
| F8 | Shifted and rotated noncontinuous Rastrigin's function | (-100, 100) | 800 | Multimodal |
| F9 | Shifted and rotated Levy function | (-100, 100) | 900 | Multimodal |
| F10 | Shifted and rotated Schwefel's function | (-100, 100) | 1000 | Multimodal |
| F11 | Hybrid function 1 (N=3) | (-100, 100) | 1100 | Hybrid |
| F12 | Hybrid function 2 (N=3) | (-100, 100) | 1200 | Hybrid |
| F13 | Hybrid function 3 (N=3) | (-100, 100) | 1300 | Hybrid |
| F14 | Hybrid function 4 (N=4) | (-100, 100) | 1400 | Hybrid |
| F15 | Hybrid function 5 (N=4) | (-100, 100) | 1500 | Hybrid |
| F16 | Hybrid function 6 (N=4) | (-100, 100) | 1600 | Hybrid |
| F17 | Hybrid function 6 (N=5) | (-100, 100) | 1700 | Hybrid |
| F18 | Hybrid function 6 (N=5) | (-100, 100) | 1800 | Hybrid |
| F19 | Hybrid function 6 (N=5) | (-100, 100) | 1900 | Hybrid |
| F20 | Hybrid function 6 (N=6) | (-100, 100) | 2000 | Hybrid |
| F21 | Composition function 1 (N=3) | (-100, 100) | 2100 | Composition |
| F22 | Composition function 2 (N=3) | (-100, 100) | 2200 | Composition |
| F23 | Composition function 3 (N=4) | (-100, 100) | 2300 | Composition |
| F24 | Composition function 4 (N=4) | (-100, 100) | 2400 | Composition |
| F25 | Composition function 5 (N=5) | (-100, 100) | 2500 | Composition |
| F26 | Composition function 6 (N=5) | (-100, 100) | 2600 | Composition |
| F27 | Composition function 7 (N=6) | (-100, 100) | 2700 | Composition |
| F28 | Composition function 8 (N=6) | (-100, 100) | 2800 | Composition |
| F29 | Composition function 9 (N=3) | (-100, 100) | 2900 | Composition |
| F30 | Composition function 10 (N=3) | (-100, 100) | 3000 | Composition |

TABLE 3: Comparison performance between AGMPSO and PSO, TSLPSO, HFPSSO and MPEPSO on CEC 2017 benchmark functions (10-D).

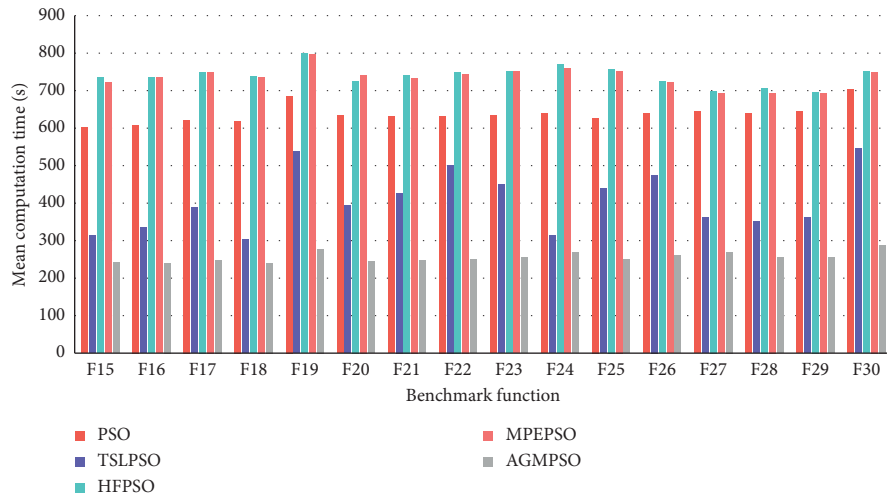
| F <i>n</i> | AGMPSO | | | TSLPSO | | | HFPSSO | | | MPEPSO | | | PSO | | |
|------------|-----------------|----------|----------|-----------------|----------|----------|-----------------|----------|----------|----------|----------|----------|-----------------|----------|----------|
| | Mean | Std | <i>h</i> | Mean | Std | <i>h</i> | Mean | Std | <i>h</i> | Mean | Std | <i>h</i> | Mean | Std | <i>h</i> |
| F1 | 1.02E+03 | 1.94E+03 | + | 2.53E+03 | 2.06E+03 | + | 6.03E+08 | 2.83E+08 | + | 3.82E+10 | 2.79E+08 | + | 4.86E+08 | 1.76E+08 | + |
| F3 | 3.00E+02 | 5.46E+01 | - | 3.00E+02 | 1.16E+01 | - | 3.78E+03 | 1.62E+03 | + | 3.29E+03 | 3.78E+03 | + | 8.02E+03 | 5.18E+03 | + |
| F4 | 4.06E+02 | 1.37E+01 | - | 4.83E+02 | 5.09E+01 | - | 4.47E+02 | 3.70E+01 | + | 5.44E+03 | 4.46E+01 | + | 4.16E+02 | 1.93E+01 | + |
| F5 | 5.44E+02 | 1.83E+01 | + | 5.94E+02 | 3.36E+00 | + | 5.83E+02 | 1.50E+01 | + | 6.91E+02 | 6.36E+00 | + | 5.79E+02 | 2.46E+01 | + |
| F6 | 6.19E+02 | 9.40E+00 | + | 6.56E+02 | 8.10E+00 | + | 6.11E+02 | 8.77E+00 | - | 6.76E+02 | 7.65E+00 | + | 6.40E+02 | 1.32E+01 | + |
| F7 | 7.36E+02 | 1.13E+01 | + | 8.15E+02 | 3.21E+00 | + | 7.40E+02 | 7.74E+00 | + | 8.29E+02 | 4.04E+00 | + | 7.77E+02 | 2.14E+01 | + |
| F8 | 8.24E+02 | 9.43E+00 | + | 1.00E+03 | 2.54E+00 | + | 8.25E+02 | 4.87E+00 | + | 9.05E+02 | 4.87E+00 | + | 8.60E+02 | 9.96E+00 | + |
| F9 | 9.56E+02 | 1.07E+02 | + | 9.98E+02 | 3.19E+02 | + | 1.12E+03 | 1.13E+02 | + | 1.73E+03 | 2.95E+01 | + | 1.33E+03 | 3.41E+02 | + |
| F10 | 2.09E+03 | 3.46E+02 | - | 1.64E+03 | 1.10E+02 | - | 2.11E+03 | 2.35E+02 | - | 4.56E+03 | 7.17E+02 | + | 2.85E+03 | 2.15E+02 | + |
| F11 | 1.13E+03 | 1.85E+01 | - | 1.30E+03 | 1.97E+00 | - | 1.11E+03 | 9.71E+00 | - | 2.51E+03 | 1.05E+02 | + | 1.57E+03 | 9.37E+02 | + |
| F12 | 5.33E+03 | 9.64E+03 | + | 1.41E+04 | 1.25E+04 | + | 8.13E+05 | 5.31E+05 | + | 4.75E+09 | 9.05E+08 | + | 7.22E+07 | 2.03E+08 | + |
| F13 | 1.64E+03 | 2.42E+01 | + | 1.85E+03 | 1.51E+02 | + | 6.17E+03 | 2.34E+03 | + | 5.26E+08 | 9.70E+08 | + | 3.11E+06 | 4.30E+06 | + |
| F14 | 1.46E+03 | 2.75E+00 | + | 1.60E+03 | 4.40E+00 | + | 4.01E+03 | 1.94E+02 | + | 1.84E+04 | 1.51E+04 | + | 6.01E+03 | 4.10E+03 | + |
| F15 | 1.58E+03 | 6.96E+01 | - | 1.59E+03 | 1.86E+00 | - | 1.71E+03 | 1.94E+02 | + | 8.53E+03 | 3.32E+03 | + | 1.71E+04 | 6.12E+03 | + |
| F16 | 1.86E+03 | 1.05E+02 | + | 1.60E+03 | 2.19E+01 | + | 1.96E+03 | 2.85E+01 | + | 3.10E+03 | 1.09E+02 | + | 2.12E+03 | 1.87E+02 | + |
| F17 | 1.77E+03 | 3.23E+01 | + | 1.89E+03 | 4.12E+00 | + | 1.72E+03 | 1.12E+01 | - | 2.79E+03 | 2.59E+02 | + | 1.83E+03 | 4.76E+01 | + |
| F18 | 4.96E+03 | 8.96E+03 | + | 3.59E+04 | 6.25E+02 | + | 1.87E+03 | 6.90E+01 | - | 1.19E+10 | 3.29E+09 | + | 1.85E+07 | 2.77E+07 | + |
| F19 | 4.00E+03 | 7.86E+03 | + | 5.90E+03 | 7.03E+01 | + | 1.94E+03 | 6.64E+01 | - | 1.11E+10 | 5.21E+08 | + | 2.04E+05 | 5.53E+05 | + |
| F20 | 2.16E+03 | 3.22E+01 | - | 2.00E+03 | 3.77E+01 | - | 2.10E+03 | 5.33E+01 | - | 2.56E+03 | 1.02E+02 | + | 2.23E+03 | 8.20E+01 | + |
| F21 | 2.30E+03 | 5.84E+01 | + | 2.70E+03 | 3.91E+01 | + | 2.24E+03 | 5.69E+01 | - | 2.76E+03 | 6.96E+00 | + | 2.36E+03 | 5.40E+01 | + |
| F22 | 2.33E+03 | 1.45E+01 | + | 2.64E+03 | 2.98E+01 | + | 2.40E+03 | 5.86E+01 | - | 4.91E+03 | 2.81E+01 | + | 2.37E+03 | 2.10E+02 | + |
| F23 | 2.67E+03 | 3.40E+01 | - | 2.70E+03 | 3.48E+00 | - | 2.94E+03 | 1.65E+02 | + | 3.92E+03 | 8.02E+01 | + | 2.73E+03 | 8.73E+01 | + |
| F24 | 2.70E+03 | 1.39E+02 | + | 2.74E+03 | 8.46E+01 | - | 2.64E+03 | 3.51E+01 | - | 3.36E+03 | 5.65E+00 | + | 2.78E+03 | 1.25E+02 | + |
| F25 | 2.92E+03 | 2.17E+00 | + | 2.98E+03 | 5.37E+01 | - | 2.96E+03 | 1.13E+01 | + | 4.67E+03 | 1.90E+01 | + | 2.99E+03 | 1.47E+02 | + |
| F26 | 3.13E+03 | 4.72E+02 | + | 3.80E+03 | 1.43E+02 | + | 3.56E+03 | 3.12E+02 | + | 5.51E+03 | 5.00E+01 | + | 3.53E+03 | 6.80E+02 | + |
| F27 | 3.13E+03 | 3.73E+01 | + | 3.18E+03 | 1.09E+00 | - | 3.38E+03 | 1.34E+02 | + | 4.66E+03 | 9.29E+01 | + | 3.19E+03 | 7.89E+01 | + |
| F28 | 3.32E+03 | 9.61E+00 | + | 3.78E+03 | 6.29E+01 | + | 3.49E+03 | 2.48E+02 | + | 4.40E+03 | 2.51E+01 | + | 3.32E+03 | 1.20E+02 | - |
| F29 | 3.24E+03 | 6.01E+00 | - | 3.34E+03 | 1.03E+01 | - | 3.24E+03 | 2.95E+01 | + | 3.27E+03 | 3.86E+01 | + | 3.46E+03 | 1.29E+02 | + |
| F30 | 3.45E+05 | 7.06E+05 | + | 7.61E+05 | 3.15E+03 | + | 1.23E+05 | 5.21E+04 | + | 4.32E+08 | 1.00E+07 | + | 1.23E+07 | 1.65E+07 | + |
| Scores | 14 | | | 4 | | | 10 | | | 0 | | | 1 | | |
| Rank | 1 | | | 3 | | | 2 | | | 5 | | | 4 | | |

TABLE 4: Comparison performance between AGMPSO and PSO, TSLPSO, HFPSSO and MPEPSO on CEC 2017 benchmark functions (30-D).

| F <i>n</i> | AGMPSO | | | TSLPSO | | | HFPSSO | | | MPEPSO | | | PSO | | |
|------------|-----------------|----------|----------|-----------------|----------|----------|-----------------|----------|----------|-----------------|----------|----------|-----------------|----------|----------|
| | Mean | Std | <i>h</i> | Mean | Std | <i>h</i> | Mean | Std | <i>h</i> | Mean | Std | <i>h</i> | Mean | Std | <i>h</i> |
| F1 | 1.03E+03 | 1.64E+03 | + | 2.41E+03 | 1.86E+03 | + | 6.27E+08 | 2.33E+08 | + | 2.62E+10 | 2.49E+08 | + | 5.20E+08 | 1.48E+08 | + |
| F3 | 3.00E+02 | 4.81E+04 | - | 3.00E+02 | 9.91E+03 | - | 4.12E+03 | 1.44E+03 | + | 3.33E+03 | 3.41E+03 | + | 7.71E+03 | 4.25E+03 | + |
| F4 | 4.39E+02 | 1.15E+01 | - | 4.79E+02 | 4.53E+01 | - | 4.57E+02 | 2.97E+01 | + | 5.67E+03 | 3.57E+01 | + | 4.41E+02 | 1.59E+01 | + |
| F5 | 5.66E+02 | 1.49E+01 | + | 6.00E+02 | 2.76E+00 | + | 6.01E+02 | 1.36E+01 | + | 7.61E+02 | 5.47E+00 | + | 6.08E+02 | 2.14E+01 | + |
| F6 | 6.50E+02 | 7.61E+00 | + | 6.32E+02 | 9.91E+00 | + | 6.29E+02 | 7.63E+00 | - | 6.16E+02 | 6.59E+00 | - | 6.60E+02 | 1.09E+01 | + |
| F7 | 7.46E+02 | 9.29E+00 | + | 7.67E+02 | 2.73E+00 | + | 8.15E+02 | 6.89E+00 | + | 8.87E+02 | 3.52E+00 | + | 7.08E+02 | 1.80E+01 | + |
| F8 | 8.03E+02 | 1.55E+00 | + | 1.06E+03 | 2.24E+00 | + | 8.66E+02 | 3.58E+00 | + | 8.24E+02 | 4.09E+00 | + | 8.91E+02 | 8.75E+00 | + |
| F9 | 1.04E+03 | 9.31E+01 | N/A | N/A | N/A | N/A | 1.06E+03 | 1.02E+02 | + | 1.86E+03 | 2.63E+01 | + | 1.42E+03 | 2.97E+02 | + |
| F10 | 2.14E+03 | 2.84E+02 | - | 1.76E+03 | 9.87E+01 | - | 2.05E+03 | 2.12E+02 | + | 4.89E+03 | 6.32E+02 | + | 2.97E+03 | 1.92E+02 | + |
| F11 | 1.11E+03 | 1.47E+00 | - | 1.41E+03 | 1.62E+00 | - | 1.12E+03 | 8.45E+00 | - | 2.47E+03 | 9.17E+02 | + | 1.51E+03 | 7.59E+02 | + |
| F12 | 5.19E+03 | 7.91E+03 | + | 1.30E+04 | 1.00E+04 | + | 7.65E+05 | 4.30E+05 | + | 4.71E+09 | 7.51E+08 | + | 7.80E+07 | 1.71E+08 | + |
| F13 | 1.56E+03 | 2.09E+02 | + | 1.67E+03 | 1.26E+02 | + | 6.17E+03 | 2.04E+03 | + | 5.59E+08 | 8.64E+08 | + | 2.81E+06 | 3.58E+06 | + |
| F14 | 1.43E+03 | 2.31E+01 | + | 1.57E+03 | 3.96E+00 | + | 4.37E+03 | 1.42E+03 | + | 1.75E+04 | 1.31E+04 | + | 5.82E+03 | 3.61E+03 | + |
| F15 | 1.54E+03 | 5.92E+01 | - | 1.58E+03 | 1.56E+00 | - | 1.60E+03 | 1.71E+02 | + | 9.22E+03 | 2.70E+03 | + | 1.64E+04 | 5.21E+03 | + |
| F16 | 1.63E+03 | 1.16E+01 | + | 1.67E+03 | 1.82E+01 | + | 1.97E+03 | 2.37E+01 | + | 3.11E+03 | 9.40E+01 | + | 2.06E+03 | 1.52E+02 | + |
| F17 | 1.73E+03 | 2.72E+00 | + | 1.71E+03 | 3.30E+00 | + | 1.70E+03 | 9.58E+00 | - | 2.80E+03 | 2.26E+02 | + | 1.91E+03 | 4.14E+01 | + |
| F18 | 4.47E+03 | 7.71E+03 | + | 3.33E+04 | 5.07E+02 | + | 1.85E+03 | 5.66E+01 | - | 1.12E+10 | 2.96E+09 | + | 1.78E+07 | 2.49E+07 | + |
| F19 | 4.28E+03 | 6.69E+03 | + | 6.37E+03 | 5.63E+01 | + | 1.90E+03 | 5.35E+01 | - | 1.15E+10 | 4.38E+08 | + | 1.96E+05 | 4.59E+05 | + |
| F20 | 2.00E+03 | 2.48E+01 | + | 2.01E+03 | 3.32E+01 | + | 2.09E+03 | 4.53E+01 | - | 2.49E+03 | 9.05E+01 | + | 2.42E+03 | 7.22E+01 | + |
| F21 | 2.26E+03 | 4.67E+01 | + | 2.79E+03 | 3.53E+01 | + | 2.33E+03 | 4.67E+01 | - | 3.01E+03 | 5.92E+00 | + | 2.29E+03 | 4.38E+01 | + |
| F22 | 2.24E+03 | 1.31E+02 | + | 2.73E+03 | 2.48E+01 | + | 2.29E+03 | 4.99E+01 | - | 4.42E+03 | 2.47E+01 | + | 2.85E+03 | 2.10E+02 | + |
| F23 | 2.58E+03 | 2.76E+00 | + | 2.61E+03 | 3.14E+00 | - | 2.89E+03 | 1.46E+02 | + | 4.08E+03 | 6.82E+01 | + | 2.67E+03 | 7.16E+01 | + |
| F24 | 2.63E+03 | 1.20E+02 | + | 3.70E+03 | 7.53E+01 | - | 2.60E+03 | 3.02E+01 | + | 2.59E+03 | 4.70E+00 | + | 2.73E+03 | 1.03E+02 | + |
| F25 | 2.85E+03 | 1.94E+01 | - | 2.87E+03 | 4.35E+01 | - | 2.91E+03 | 9.73E+00 | + | 4.40E+03 | 1.62E+01 | + | 2.95E+03 | 1.21E+02 | + |
| F26 | 3.10E+03 | 3.97E+01 | + | 3.72E+03 | 1.22E+02 | + | 3.35E+03 | 2.53E+02 | + | 5.57E+03 | 4.20E+01 | + | 3.32E+03 | 5.71E+02 | + |
| F27 | 3.09E+03 | 3.33E+01 | + | 3.13E+03 | 9.10E+01 | - | 3.29E+03 | 1.07E+02 | + | 4.48E+03 | 8.27E+01 | + | 3.13E+03 | 6.87E+01 | + |
| F28 | 3.26E+03 | 8.27E+01 | + | 3.70E+03 | 5.54E+01 | + | 3.46E+03 | 2.06E+02 | + | 4.28E+03 | 2.16E+01 | + | 3.25E+03 | 1.02E+02 | - |
| F29 | 2.96E+03 | 4.99E+01 | + | 3.25E+03 | 8.25E+00 | + | 3.08E+03 | 2.54E+01 | + | 3.60E+03 | 3.09E+01 | + | 3.25E+03 | 1.10E+02 | + |
| F30 | 3.25E+05 | 5.65E+05 | + | 7.00E+05 | 2.84E+03 | + | 8.96E+04 | 4.53E+04 | + | 4.21E+08 | 8.41E+06 | + | 1.15E+07 | 1.32E+07 | + |
| Scores | 19 | | | 2 | | | 5 | | | 2 | | | 1 | | |
| Rank | 1 | | | 4 | | | 2 | | | 3 | | | 5 | | |

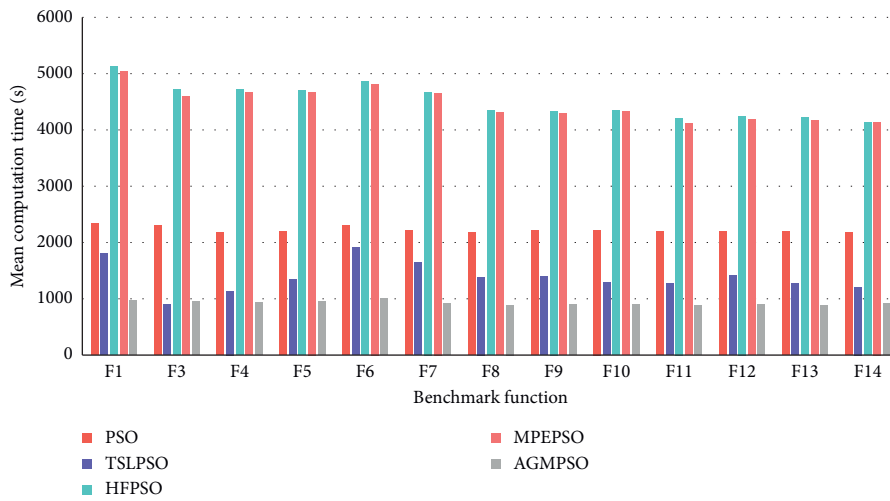


(a)



(b)

FIGURE 4: The average computation time of comparison algorithms on benchmark functions (10-D).



(a)

FIGURE 5: Continued.

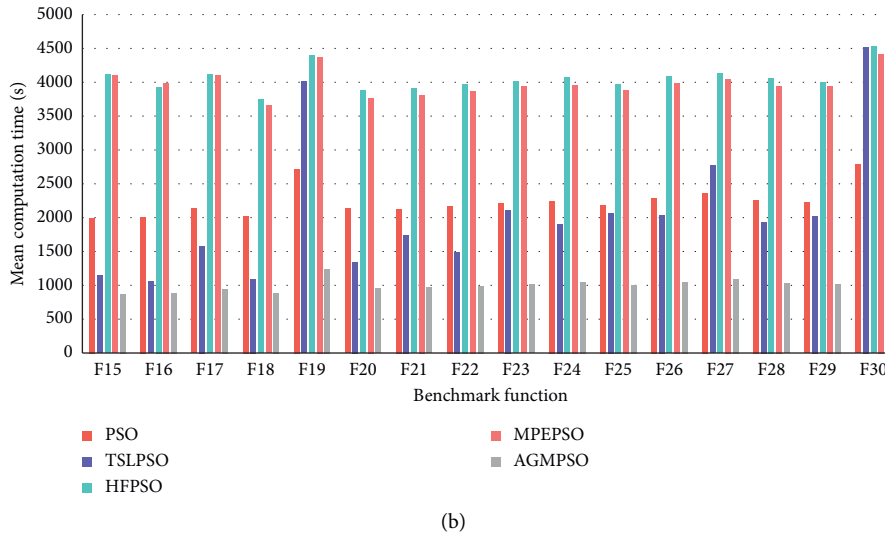


FIGURE 5: The average computation time of comparison algorithms on benchmark functions (30-D).

fitness values equal those of winning algorithms, which makes AGMPSO take the second rank on the above-mentioned function.

For 30-dimensional experiments in Table 4, since the iterations greatly increase, the results of all algorithms have improved in varying degrees. AGMPSO achieves optimal mean fitness value on F3 and F20. It gets the same performance as in 10-dimensional experiments in unimodal and multimodal functions, gets 7 out of 10 best results in hybrid functions, and 6 out of 10 best results in composition functions. In general, AGMPSO outperforms peers for the 19 benchmark functions.

In short, AGMPSO is top ranked in both 10-dimensional and 30-dimensional experiments.

The standard deviations of AGMPSO on different benchmark functions are generally smaller than those of comparison PSO variants, which indicates better robustness of our algorithm.

Aiming to analyze the computational efficiency of compared peers, the average computation time for each algorithm to run all benchmark functions is depicted in Figures 4 and 5. It can be concluded that AGMPSO consumes lower computational overhead than its peers and the advantage grows obvious when iteration increases as shown in Figure 5. It is worth noting that, despite shared the same time complexity of standard PSO, the time consumption of AGMPSO is significantly lower indicating higher computational efficiency.

In summary, the outstanding results of AGMPSO in terms of average computational time and fitness values demonstrate that our proposed algorithm obtains higher search accuracy and convergence rate than its peers, meanwhile, with the significant robust.

3.2.3. Wilcoxon's Rank Sum Test Results. The Wilcoxon rank sum test at a significance level of $\alpha = 0.05$ is performed on the ranking between AGMPSO and other PSO variants to

analyze their statistical significance. Tables 3 and 4 list the results of Wilcoxon rank sum test on fitness values of all 29 functions. In both tables, h value (+/-/~) indicates that the AGMPSO performs significantly better, significantly worse, or not statistically significant than its competitor.

It can be observed from the results that AGMPSO is significantly better than compared PSO variants in most of test functions. AGMPSO gets the same mean values with TSLPSO, standard PSO, and HFPSO on 10-dimensional F3, F28, and F29, while with TSLPSO on 30-dimensional F3. However, from the rank sum test results, AGMPSO performs significantly worse than the peer on above functions. Although the proposed algorithm does not gain the superior rank on these functions, it still takes the second rank in mean fitness value on each function, and the difference is slight.

3.2.4. Convergence Progresses. In order to observe the convergence speed of all the peer algorithms, the convergence processes in random runs of the comparison algorithms on benchmark functions of 10 dimensions are depicted in Figures 6 and 7.

In Figure 6, AGMPSO does not show the characteristics of rapid convergence on F1 and F3 in the initial period because of the strong particle diversity yielded by proposed mutation strategy; meanwhile, the high solution accuracy is achieved by our algorithm at the later convergence stage.

From F4 to F10, we can observe that the other comparison algorithms fall into the local minima to different extent, while AGMPSO attains the favorable performances on all multimodal functions. However, it is noteworthy that the effect of proposed adaptive strategies is not obvious on F4, F6, and F9, which may cause the low diversity in the early period and fail to find the global optima. This is demonstrated by the result of F6 in Table 3.

From the performances for F11 to F20, the similar problems of rapid convergence in the initial period can be observed on F11, F14, and F15. Furthermore, AGMPSO fails

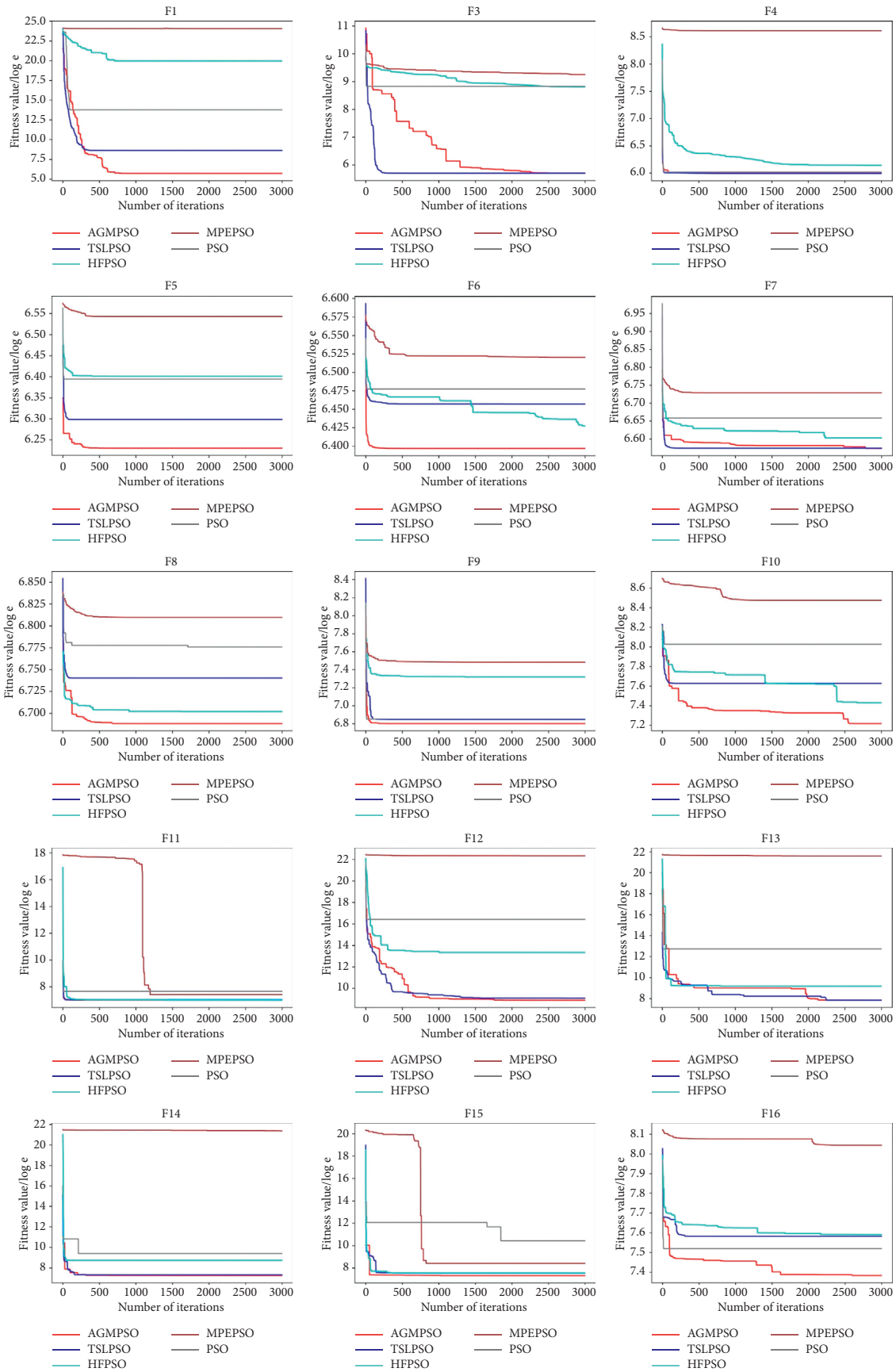


FIGURE 6: Convergence curves of comparison algorithms on F1–F16.

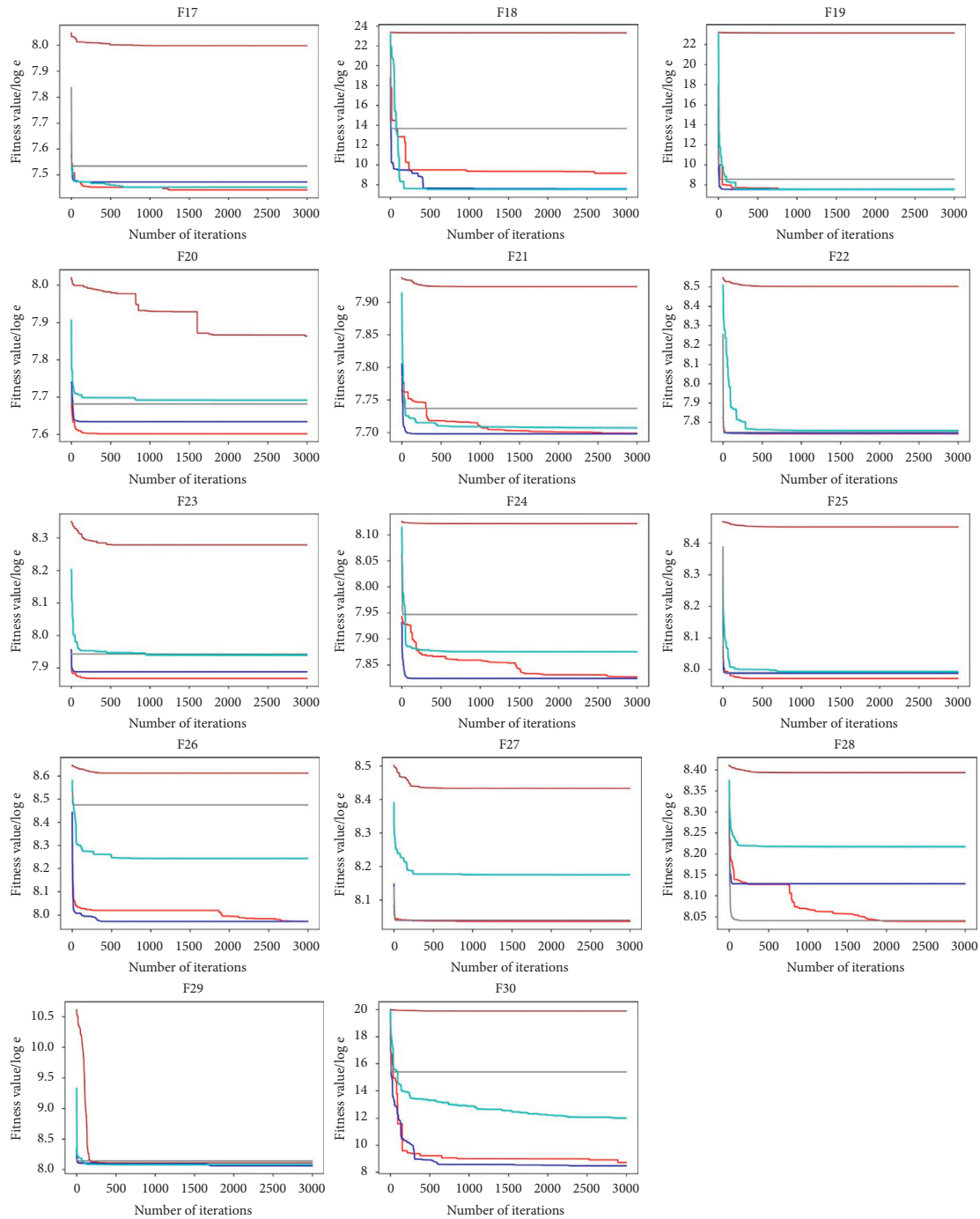


FIGURE 7: Convergence curves of comparison algorithms on F17–F30.

to achieve the satisfactory results on F18. Despite with the 4 outstanding achievements and the improvement in 30-dimensional experiment results, due to the rapid convergence rate at the early stage, the exploration capability of proposed algorithm in solving hybrid functions should be promoted.

The results presented in F21 to F30 depict that exploitation capability of AGMPSO is higher than most of the peers. The gradual convergence process on F21, F24, F26, F28, and F30 can be seen, which reflects the balance of exploration and exploitation of our algorithm.

The conscious summary can be drawn that AGMPSO performs well on most of functions. Meanwhile, the proposed algorithm does not yield satisfactory performances on hybrid functions, so do the other PSO variants, which suggests much room for improvement in solving these functions. It turns out from the 30-dimensional comparison results that, despite the low efficiency, increasing the number of iterations could be an improvement insight worth discussing.

By contrast with the other competitors, a relatively slow and gradual convergence curve presents in solving many

functions, which exactly portrays the algorithm's intent of finding out more promising solutions.

4. Conclusion

Adaptive particle swarm optimization with Gaussian perturbation and mutation is proposed to address the existing drawback of standard PSO. To prevent trapping into local optimum, Gaussian perturbation is implemented to global optima, further increasing the exploitation capability. For the nonoptimal particles that fall into the evolutionary stagnation, the mutation is leveraged to promote the particles' diversity and utilization to improve the exploration ability. Simultaneously, the adaptive strategy regulates the interference level of Gaussian perturbation and mutation during different evolutionary stages in order to balance the searching ability and accuracy. The visual result of aggregation analysis validates this dynamic process. The performance on benchmark functions of CEC 2017 test suits manifests that AGMPSO outperforms its competitors by a big margin in terms of searching accuracy, searching reliability, and searching efficiency.

In future works, considering the powerful global search ability of the particle swarm algorithm, it can be considered to optimize the topology, connection weights, and thresholds of the neural networks or combine the global optimization ability of PSO with the local optimization ability of the BPNN to improve the generalization and learning performance of the neural network.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grants 71371181 and 71672193 and by the Research Foundation of Xian International Studies University under grant BSZA2019003.

References

- [1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, Pretoria, South Africa, June 2002.
- [2] W. Han, P. Yang, H. Ren, and J. Sun, "Comparison study of several kinds of inertia weights for PSO," in *Proceedings of the 2010 IEEE International Conference on Progress in Informatics and Computing*, vol. 1, pp. 280–284, IEEE, Beijing, China, December 2010.
- [3] J. C. Xu, T. H. Xu, L. Sun, and J. Y. Ren, "Feature selection for cancer classification based on neighborhood rough set and particle swarm optimization," *Journal of Chinese Computer Systems*, vol. 35, no. 11, pp. 2528–2532, 2014.
- [4] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.
- [5] Z. H. Zhan, J. Zhang, Y. Li, and S. H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics*, vol. 39, no. 6, pp. 1362–1381, 2009.
- [6] R. Cheng and Y. Jin, "A social learning particle swarm optimization algorithm for scalable optimization," *Information Sciences*, vol. 291, pp. 43–60, 2015.
- [7] B. Alatas, E. Akin, and A. B. Ozer, "Chaos embedded particle swarm optimization algorithms," *Chaos, Solitons & Fractals*, vol. 40, no. 4, pp. 1715–1734, 2009.
- [8] X. C. Zhao, G. L. Liu, H. Q. Liu, and G. S. Zhao, "Particle swarm optimization algorithm based on non-uniform mutation and multiple stages perturbation," *Chinese Journal of Computers*, vol. 60, pp. 1–20, 2014.
- [9] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.
- [10] W.-N. Chen, J. Zhang, Y. Lin et al., "Particle swarm optimization with an aging leader and challengers," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 2, pp. 241–258, 2013.
- [11] X. U. Xiao-Bo, K. F. Zheng, L. I. Dan, W. U. Bin, and Y. X. Yang, "New chaos-particle swarm optimization algorithm," *Journal on Communications*, vol. 33, no. 1, pp. 24–16, 2012.
- [12] V. D. B. Frans and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.
- [13] D. Tian, "Particle swarm optimization with chaos-based initialization for numerical optimization," *Intelligent Automation & Soft Computing*, vol. 24, no. 2, pp. 331–342, 2017.
- [14] W. B. Du, W. Ying, G. Yan, Y. B. Zhu, and X. B. Cao, "Heterogeneous strategy particle swarm optimization," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 4, pp. 467–471, 2016.
- [15] M. Munlin and M. Anantathanavit, "Hybrid radius particle swarm optimization," in *Proceedings of the 2016 IEEE Region 10 Conference (TENCON)*, pp. 2180–2184, IEEE, Singapore, November 2016.
- [16] M. S. Kiran, "Particle swarm optimization with a new update mechanism," *Applied Soft Computing*, vol. 60, pp. 670–678, 2017.
- [17] K. Elumalai, M. Elumalai, K. Eluri et al., "Facile synthesis, spectral characterization, antimicrobial and in vitro cytotoxicity of novel N3, N5-diisonicotinyl-2, 6-dimethyl-4-phenyl-1, 4-dihydropyridine-3, 5-dicarbohydrazide derivatives," *Bulletin of Faculty of Pharmacy, Cairo University*, vol. 1, no. 54, pp. 77–86, 2016.
- [18] S. Rastegar, R. Araújo, and J. Mendes, "Online identification of Takagi-Sugeno fuzzy models based on self-adaptive hierarchical particle swarm optimization algorithm," *Applied Mathematical Modelling*, vol. 45, pp. 606–620, 2017.
- [19] A. A. Nagra, F. Han, and Q. H. Ling, "An improved hybrid self-inertia weight adaptive particle swarm optimization algorithm with local search," *Engineering Optimization*, vol. 51, no. 7, pp. 1115–1132, 2019.
- [20] A. A. Nagra, F. Han, Q. H. Ling et al., "Hybrid self-inertia weight adaptive particle swarm optimisation with local search using C4.5 decision tree classifier for feature selection problems," *Connection Science*, vol. 32, no. 1, pp. 16–36, 2020.

- [21] Y. Xue, B. Xue, and M. Zhang, "Self-adaptive particle swarm optimization for large-scale feature selection in classification," *ACM Transactions on Knowledge Discovery from Data*, vol. 13, no. 5, pp. 1–27, 2019.
- [22] R. S. Kumar, K. Kondapaneni, V. Dixit, A. Goswami, L. S. Thakur, and M. K. Tiwari, "Multi-objective modeling of production and pollution routing problem with time window: a self-learning particle swarm optimization approach," *Computers & Industrial Engineering*, vol. 99, pp. 29–40, 2016.
- [23] M.-C. Chen, Y.-H. Hsiao, R. Himadeep Reddy, and M. K. Tiwari, "The self-learning particle swarm optimization approach for routing pickup and delivery of multiple products with material handling in multiple cross-docks," *Transportation Research Part E: Logistics and Transportation Review*, vol. 91, pp. 208–226, 2016.
- [24] M. Hu, T. Wu, and J. D. Weir, "An adaptive particle swarm optimization with multiple adaptive methods," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 705–720, 2012.
- [25] F. Wang, H. Zhang, K. Li, Z. Lin, J. Yang, and X.-L. Shen, "A hybrid particle swarm optimization algorithm using adaptive learning strategy," *Information Sciences*, vol. 436–437, pp. 162–177, 2018.
- [26] S.-F. Li and C.-Y. Cheng, "Particle swarm optimization with fitness adjustment parameters," *Computers & Industrial Engineering*, vol. 113, pp. 831–841, 2017.
- [27] X. Hu and R. C. Eberhart, "Adaptive particle swarm optimization: detection and response to dynamic systems," in *Proceedings of the 2002 Congress on Evolutionary Computation CEC'02*, vol. 2, pp. 1666–1670, Honolulu, HI, USA, May 2002.
- [28] X. F. Xie, W. J. Zhang, and Z. L. Yang, "Adaptive particle swarm optimization on individual level," vol. 2, pp. 1215–1218, in *Proceedings of the 6th International Conference on Signal Processing*, vol. 2, pp. 1215–1218, IEEE, Beijing, China, August 2002.
- [29] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, and M. Ventresca, "Enhancing particle swarm optimization using generalized opposition-based learning," *Information Sciences*, vol. 181, no. 20, pp. 4699–4714, 2011.
- [30] M. J. Mahmoodabadi, Z. Salahshoor Mottaghi, and A. Bagheri, "Hepso: high exploration particle swarm optimization," *Information Sciences*, vol. 273, no. 18, pp. 101–111, 2014.
- [31] H. Wang, H. Sun, C. Li, S. Rahnamayan, and J.-S. Pan, "Diversity enhanced particle swarm optimization with neighborhood search," *Information Sciences*, vol. 223, pp. 119–135, 2013.
- [32] Y. V. Pehlivanoglu, "A new particle swarm optimization method enhanced with a periodic mutation strategy and neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 436–452, 2012.
- [33] S. Shao, Y. Peng, C. He, and Y. Du, "Efficient path planning for UAV formation via comprehensively improved particle swarm optimization," *ISA Transactions*, vol. 97, pp. 415–430, 2020.
- [34] X. Tao, W. Guo, Q. Li, C. Ren, and R. Liu, "Multiple scale self-adaptive cooperation mutation strategy-based particle swarm optimization," *Applied Soft Computing*, vol. 89, pp. 106–124, 2020.
- [35] G. Xu, Q. Cui, X. Shi et al., "Particle swarm optimization based on dimensional learning strategy," *Swarm and Evolutionary Computation*, vol. 45, pp. 33–51, 2019.
- [36] İ. B. Aydilek, "A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems," *Applied Soft Computing*, vol. 66, pp. 232–249, 2018.
- [37] Z. Liu and T. Nishi, "Multipopulation ensemble particle swarm optimizer for engineering design problems," *Mathematical Problems in Engineering*, vol. 2020, Article ID 1450985, 30 pages, 2020.
- [38] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, *Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Bound Constrained Teal-Parameter Numerical Optimization*, Nan- yang Technological University, Singapore, 2016.

Research Article

Continuous Trust Evaluation of Power Equipment and Users Based on Risk Measurement

Congcong Shi ^{1,2,3}, Jiaxuan Fei,^{2,3} Xiaojian Zhang,^{2,3} Qigui Yao,^{2,3} and Jie Fan^{2,3}

¹State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210003, China

²Global Energy Interconnection Research Institute Co. Ltd., Nanjing 210003, China

³State Grid Key Laboratory of Information & Network Security, Nanjing 210003, China

Correspondence should be addressed to Congcong Shi; 765734893@qq.com

Received 22 July 2020; Revised 10 November 2020; Accepted 28 November 2020; Published 11 December 2020

Academic Editor: Ting Yang

Copyright © 2020 Congcong Shi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In power Internet of Things environment, the existing border-based protection system and the “one-time authentication, one-time authorization, and long-term effective” approach are difficult to deal with the threat of attacks from internal and external devices and users with legal authority. In order to solve the problem of authorized access of power equipment and users, combined with behavior risk assessment, a continuous trust evaluation scheme of power equipment and users is presented in this paper. The scheme is evaluated by the combination of direct trust, indirect trust, and comprehensive trust and adds the penalty reward factor and time attenuation function to improve the reliability of the results. In addition, this paper will quantify the risk of the behavior of power equipment and users and regard it as a factor affecting the degree of trust, so as to achieve continuous trust evaluation of equipment and users.

1. Introduction

1.1. Background. The Internet of Things technology mainly relies on related sensing equipment to connect objects to the network according to an agreed protocol. In the power system, the use of the Internet of Things technology can better control power equipments, power personnel, and the operating environment, specifically in the four aspects of perception, identification, interconnection, and control. Through the power Internet of Things technology, the operating efficiency of the power system can be greatly improved. For example, smart meters can upload user-side data to the power grid company through the network to avoid manual copying of the wrong meters. By connecting the power station to the power system, the power Internet of Things can be used to achieve dispatch control.

In the power Internet of things environment, with the extensive access of massive terminal equipment and users, the network exposure increases, which brings severe challenges to the existing protection system. However, the existing authentication and access control for IoT terminal

equipment and users mostly adopts the method of “once authentication, once authorization, and long-term effectiveness.” After the authentication is passed, it has legal authority for a long time and can carry out any operation within the scope of authority. Due to the lack of continuous behavior analysis and authentication and access control measures, it is impossible to solve the problem that legitimate terminal devices or users are illegally controlled by attackers and access company data and business resources in a legal capacity. At the same time, for the insiders, due to the preset trust mechanism for the insiders if the insiders carry out illegal operations or launch malicious attacks, it is difficult to effectively control and will cause huge losses.

It is difficult to meet such security requirements only by relying on the traditional security architecture based on border protection. The core idea of the zero-trust architecture is that no person, device, or system inside and outside the network should be trusted by default, and the trust basis of access control should be reconstructed based on authentication and authorization. It means a never trust and always authenticate security model. In the zero-trust

architecture mode, it can well solve the problem of internal personnel violations or malicious attacks and provide guarantee for the realization of power Internet of things “any time, any place, any person, and any thing” information connection and secure interaction [1].

Zero-trust architecture needs to study continuous identity authentication and trust evaluation, through real-time evaluation of the trust of devices and users, adjust the authority level of users, and achieve accurate management and control. In order to understand the problem of trust evaluation calculation, this paper proposes a power Internet of things equipment and user trust evaluation scheme based on risk measurement. The general trust calculation does not take into account the impact of behavioral risk factors on trust. In this paper, the behavior risk value is added to the trust degree calculation, and it is calculated as a part of the trust degree calculation by quantifying the behavior risk value of power equipment and users. In addition, when calculating the trust degree, the dynamic adaptability of the calculation and the ability of the system to resist malicious attacks are enhanced by dividing the trust degree into direct trust degree and indirect trust degree and obtaining a comprehensive trust degree.

2. Zero-Trust Model

Zero-trust architecture is an end-to-end approach to network/data security [2]. Zero trust is an architectural approach that focuses on data protection. Its focus is to restrict access to resources to those who “need to know.” The traditional security architecture focuses on border defense, and authorized users can freely access resources. There is nothing this model can do about attacks from within the network. The zero-trust protection architecture is intended to eliminate unauthorized access to data and services and to make the implementation of access control as detailed as possible [3]. To reduce uncertainty (because they cannot be completely eliminated), the focus is on authentication, authorization, and narrowing the implicit trust zone, while minimizing time delays in network authentication mechanisms. Access rules are limited to minimum permissions and are as detailed as possible. A common zero-trust architecture model is shown in Figure 1.

The key components include

- (1) Policy engine (PE): this component is responsible for the final decision on whether to grant the specified access subject access to the resource (access object). It gives the data to the trust engine to calculate the trust value.
- (2) Policy administrator (PA): this component is responsible for establishing a connection between the client and the resource (a logical responsibility, not a physical connection). It generates any authentication

tokens or credentials that the client uses to access enterprise resources. It is closely related to the policy engine and depends on its decision to eventually allow or deny the connection.

- (3) Policy enforcement point (PEP): this system is responsible for enabling, monitoring, and ultimately terminating the connection between principals and enterprise resources.

The policy engine is the core of the zero-trust architecture, which decides whether to grant access to resources according to the output of the trust algorithm. The policy engine uses external information, such as IP blacklists and threat intelligence services, as input to the trust algorithm to decide whether to grant or deny access to the resource. The policy engine is paired with the policy administrator component. The policy engine makes (and records) decisions, and the policy manager executes the decisions (approve or reject). The use of appropriate trust algorithm plays a vital role in the security protection of the whole system. In the next section of this paper, we will discuss in detail the algorithm used by the trust engine when deploying the zero-trust architecture in the power IoT system.

3. Trust Evaluation Model

Eigen Trust model [4] is a trust-based access control model, which gives more weight to users with high degree of direct trust in the process of trust calculation and believes that users with greater degree of direct trust are more trustworthy. However, this method does not take into account the subjectivity and uncertainty of trust. The penalty factor is added in Claudiu’s Model [5], which improves the dynamic adaptability of the model, which enhances the antiattack ability to some extent, but the model does not take into account the historical value, which will lead to the misoperation as an attack and lead to access failure.

Through the analysis and research of the above typical trust models, this paper fully considers the trust degree of historical interaction records in the process of trust calculation, and records are used to update trust after each interaction is completed, which is conducive to a virtuous circle of trust. Maintain a cloud environment with good services. The results are fed back in real time, which are used to update the trust degree. In addition, this paper also adds risk factors to the calculation of trust degree, which makes the calculation of trust degree more prepared and in line with the reality.

The trust evaluation model used in this article is shown in Figure 2.

The trust engine acquires the relevant information of the access device or user transmitted by the policy engine, such as the resource requested, the IP address of the access device, and the identity information of the user. This information is first used to calculate the initial trust degree. The initial trust

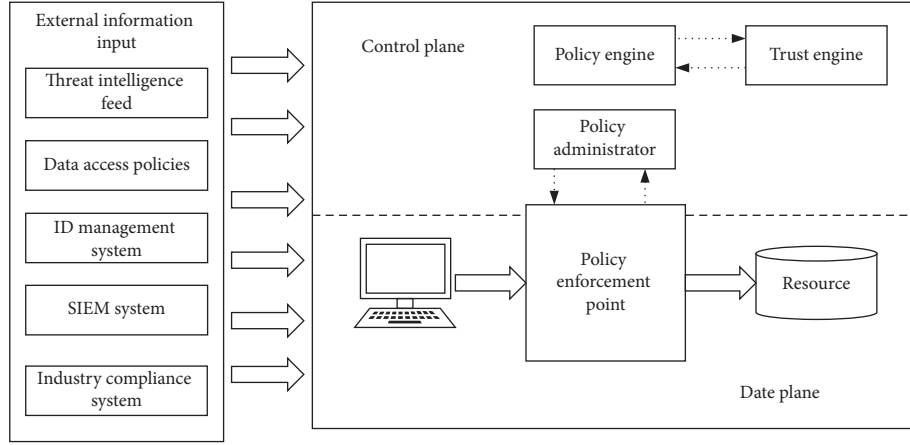


FIGURE 1: Zero-trust model.

degree is composed of direct trust degree and indirect trust degree. After the initial trust calculation is completed, the trust engine will evaluate the risk of this behavior and get a risk value, which will be used to calculate the new trust degree. Finally, the trust degree is fed back to the policy engine for subsequent access control.

3.1. Calculation of Direct Trust Degree. Direct trust degree (DT) is composed of direct experience (DE) and direct knowledge (DK) [6]. For the calculation of direct trust, there are the following formulas:

$$\begin{aligned}
 DT &= \mu DE + (1 - \mu)DK, \\
 DE &= \begin{cases} \min\left(\sum_{i=1}^n w_i \lambda_i, 1\right), \\ \max\left(\sum_{i=1}^n w_i \lambda_i, 0\right), \end{cases} \\
 w_i &= \frac{2i}{n(n+1)}, \quad (i = 1, 2, 3, \dots, n), \\
 \lambda_i &= (e_i - 1)e^{-(1/f)} + e_i \left(\frac{n-f}{n}\right)^2, \\
 DK &= \frac{n-f}{n + (sl-1)f},
 \end{aligned} \tag{1}$$

where N is the number of interactive events in the recent interval, f is the number of failed interactions, λ_i is the penalty factor, which is used to adjust the trust value when the interaction fails, and sl is the service level factor.

3.2. Calculation of Indirect Trust Degree. The indirect trust degree is mainly calculated through the transitivity of trust. According to the number of recommended paths, indirect trust value can be divided into single-path recommendation and multipath recommendation. Obviously, multipath

recommendation is more in line with the actual situation. However, it is obviously unreasonable to simply accumulate the trust values under multipath. According to the actual situation, the indirect trust degree can be calculated by applying different weights to different stages under multipath.

This paper introduces the basic model of dynamic reputation tree [7]. Through the dynamic reputation tree model, other individuals who have indirect trust relationship with the subject can be clearly constructed. At the same time, we can specify the weight of different levels according to the different levels of trust difference between the subject and the recommender. The general principle is that the closer to the subject, the greater the weight of the recommender. This kind of dynamic reputation tree can be maintained with less overhead, and the corresponding weights may be adjusted according to the importance of indirect trust individuals and subjects to achieve dynamic and convenient control.

The formula for calculating the indirect trust degree in the dynamic reputation tree is as follows:

$$IT(R_i, R_j) = \begin{cases} \sum_{k=1}^n (\omega(R_k) \times DT(R_k, R_j)) \times \frac{1}{\sum_{k=1}^n \omega(R_k)}, 0, \end{cases} \tag{2}$$

where n is the number of indirect referrals and $\omega(R_k)$ is the weight factor of presenters, which can be changed according to different levels of referrals:

$$\omega(R_k) = \begin{cases} \prod_n^l (DT(R_m, R_n)), l > 0, 1, l = 0, \end{cases} \tag{3}$$

where $DT(R_m, R_n)$ represents the direct trust value of R_m to its successor node.

3.3. Calculation of Comprehensive Trust Degree. Previously, this paper has explained the calculation method of direct and indirect trust values, and the calculation of

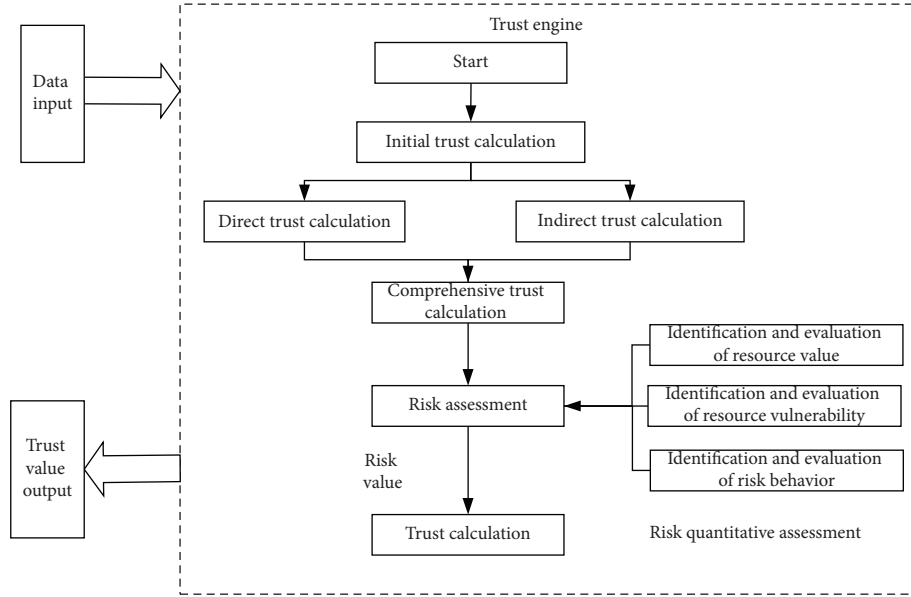


FIGURE 2: Trust computing model.

comprehensive trust value is based on the corresponding synthesis of the two values to get the user trust value at this time [8]. The specific calculation formula is as follows:

$$T(R_i, R_j) = \begin{cases} IT(R_i, R_j), & n = 0, \\ \frac{1}{1 + \beta(R_j)} \times DT(R_i, R_j) \\ + \frac{\beta(R_j)}{1 + \beta(R_j)} \times IT(R_i, R_j), & 0 < n < N, \\ DT(R_i, R_j), & n \geq N, \end{cases} \quad (4)$$

where n is the total number of historical interactive records in the system and N is the largest total number of historical interactive records in the system. $\beta(R_j)$ is the weight of direct trust and indirect trust, and it is calculated as follows:

$$\beta(R_j) = \frac{1}{2} \times [\theta(L_{R_j}) + \theta(n_{all})], \quad (5)$$

where $\theta(x) = 1 - (1/(x + \alpha))$, $\theta(L_{R_j})$ is the number of trusted referrals, and $\theta(n_{all})$ is the number of entities that have a direct trust relationship with R_j .

4. Risk Assessment Algorithm

The existing trust evaluation algorithms often use the weighted calculation method of direct trust degree and indirect trust degree [9], which will ignore the impact of user behavior risk on trust degree. This paper will quantify the user behavior risk and add it to the calculation of user trust to realize the trust evaluation of power Internet of things equipment and users based on risk measurement.

4.1. Analysis of Power Equipment and User Behavior. With the development of cloud computing, a large number of IoT devices put their services on the cloud server, which can reduce the pressure on the server and speed up the response time to a certain extent, but user behavior will also bring security risks. In this paper, the behavior of equipment terminals and users in the power things environment is divided into the following two categories:

(1) Abnormal behavior set:

The abnormal behavior set mainly refers to the fact that when the IoT terminal or user is accessing, some attributes are quite different from the usual attributes, such as landing location, accessed resources, and historical records. The details are shown in Table 1.

(2) Malicious behavior set is shown in Table 2.

5. Risk Analysis of Power Equipment and User Behavior

Combined with the definition of information security risk factors in the information security risk assessment specification, this paper defines the behavioral risk factors of IoT equipment and users in the power IoT environment as follows.

(1) Resource value (RV): the resources accessed by the equipment may be hardware resources, such as a specific watt-hour meter, or software resources, such as some data. The resource value of different levels is different. In this paper, the resource values are divided into $R = \{RV_1, RV_2, RV_3, RV_4\}$; they represent unimportant, general, important, and extremely important, respectively.

TABLE 1: Abnormal behavior set.

| |
|--------------------------|
| Behavior content |
| Location |
| IP address |
| Type of resources |
| Number of resources used |

TABLE 2: Malicious behavior set.

| |
|-----------------------------|
| Behavior content |
| SQL injection |
| Port scan |
| IP deception |
| Distributed deny attack |
| SYN flooding attack |
| Replay attack |
| Network surveillance attack |
| Virus attack |

- (2) Resource vulnerability (V): resource vulnerability refers to the difficulty in which resources are vulnerable to attack. According to the difficulty of vulnerability, resources are divided into $V = \{V_1, V_2, V_3, V_4\}$; they represent easy, ordinary, difficult, and extremely difficult, respectively.
- (3) Behavioral risk level (L): in this paper, the abnormal behavior and malicious behavior mentioned above are regarded as dangerous behavior, and the risk level of behavior is classified according to the influence degree of the behavior as $L = \{L_1, L_2, L_3, L_4\}$; they represent negligible, low, medium, and high levels of behavioral risk, respectively.

5.1. Calculation of Behavior Risk of Power Equipment and Users. Above, the behavioral risk factors of power equipment and users have been transformed into resource value R , resource vulnerability V , and behavioral risk grade L ; then, the behavioral risk assessment equation $R = RV \times V \times L$ can be obtained.

In order to participate in the calculation of trust degree later, you need to map the value at risk to the interval $[0, 1]$. The transformation formula is as follows:

$$R = \frac{\sqrt{RV \times V \times L}}{RV + V + L}. \quad (6)$$

The above formula can only statically reflect the risk level of a certain visit of the device and the user. After this, this paper introduces the dangerous behavior times c ; when the user carries on the dangerous operation continuously, the risk value should increase exponentially. In addition, the risk attenuation factor α is introduced, and the final behavioral risk assessment formula is as follows:

$$R = \begin{cases} \alpha \times R_0, & (a) \\ R_0 + \mu \times \frac{c \times \sqrt{RV \times V \times L}}{RV + V + L}. & (b) \end{cases} \quad (7)$$

In the above formula, R_0 represents the result of the most recent behavior risk calculation. When (a) represents normal behavior, the calculation of user risk value $\alpha \in [0.5, 1]$ is used to adjust the attenuation rate of user risk value. When the user behaves normally continuously, the risk value of the user attenuates. (b) represents the process of calculating the value at risk when the user has dangerous behavior, and $\mu \in [1, 2]$ is used to adjust the value at risk.

5.2. Calculation of Trust Degree of Power Equipment and Users. In this paper, based on the improved information security risk assessment equation and the Trust Model based on Behavior Risk Evolution (TMBRE), the dangerous behavior times c is introduced, and the improved calculation formula of user trust degree is obtained:

$$T = \begin{cases} \lambda^c \times T_0 + (1 - \lambda)^c \times (R - \theta), & R \in [\theta, 10], & (a) \\ T_0 + \rho^c \times (\theta - R), & R \in [0, \theta]. & (b) \end{cases} \quad (8)$$

In the above formula, θ is the threshold constant of the risk value of power equipment and user behavior. Exceeding this value means high-risk behavior. T_0 represents the user trust level that was last calculated. λ is the trust correction factor under a high-risk value, and ρ is the trust correction factor at a low-risk value. (a) is used to calculate the trust degree of power equipment and user behavior in a high-risk state. (b) is used to calculate the trust degree of power equipment and user behavior in a low-risk state.

6. Conclusion

In order to deal with the current power Internet of Things system that is difficult to deal with attacks from internal and external devices and users with legal authority, it is necessary to study continuous identity authentication, trust evaluation, and access control technologies and establish a zero-trust access control model. In this paper, a continuous trust evaluation algorithm for power IoT equipment and users based on risk measurement is

proposed, which can be used to calculate the trust degree of zero-trust architecture. Based on the analysis of the characteristics of trust, in order to enhance the dynamic adaptability and objectivity of the trust value calculation method, this paper presents a trust value calculation method with penalty factor, service level factor, and dynamic adaptation factor. In addition, this paper also adds risk factors to the calculation of trust degree, through the analysis of user behavior, quantifies the user risk behavior, and adds the number of dangerous behavior in the risk calculation; for continuous dangerous behavior, the malicious coefficient will increase exponentially.

Data Availability

The experimental data in this paper come from the actual production and operation process of the State Grid Corporation of China and are only provided on the company's internal network.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Science and Technology Project of State Grid Corporation of China (Grant no. 5700-201958466A-0-0-00): "End-to-End Security Threat Analysis and Accurate Protection of Ubiquitous Power Internet of Things."

References

- [1] R. Vanickis, P. Jacob, S. Dehghanzadeh et al., "Access control policy enforcement for zero-trust-networking," in *Proceedings of the 2018 29th Irish Signals and Systems Conference (ISSC)*, Belfast, UK, June 2018.
- [2] E. Bogner, "The zero-trust mandate: never trust, continually verify," *Software World*, vol. 50, no. 4, pp. 9-10, 2019.
- [3] A. Ghafourifar and J. K. Monroe, "Multi-party authentication in a zero-trust distributed system," US Patent 10,110,585, 2018.
- [4] D. K. Sepandar, T. S. Mario, and G. M. Hector, "The Eigen Trust algorithm for reputation management in P2P networks," in *Proceedings of the 12th International Conference on World Wide Web*, pp. 640-651, ACM Press, Budapest, Hungary, May 2003.
- [5] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: a survey," *Journal of Artificial Intelligence Research*, vol. 4, no. 1, pp. 237-285, 1996.
- [6] X. Hu, R. Jiang, M. Shi et al., "A privacy protection model for health care big data based on trust evaluation access control in cloud service environment," *Journal of Intelligent and Fuzzy Systems*, vol. 5, pp. 1-12, 2020.
- [7] T. Wang, H. Luo, W. Jia et al., "MTES: an intelligent trust evaluation scheme in sensor-cloud-enabled industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2054-2062, 2020.
- [8] R. Zhang, X. Wu, S.-Y. Zhou, and X.-S. Dong, "A trust model based on entity behavior risk assessment," *Journal of Computer Science*, vol. 32, no. 4, pp. 688-698, 2009.

- [9] J. Chen, Z. Tian, X. Cui, L. Yin, and X. Wang, "Trust architecture and reputation evaluation for internet of things," *Journal of Ambient Intelligence & Humanized Computing*, vol. 10, pp. 3099-3107, 2019.

Research Article

Intelligent Detection and Recovery of Missing Electric Load Data Based on Cascaded Convolutional Autoencoders

Xin Wang,¹ Yuanyi Chen ,² Wei Ruan ,³ Qiang Gao,¹ Guode Ying,¹ and Li Dong⁴

¹State Grid Zhejiang Electric Power Co., Ltd., Taizhou Power Supply Company, Taizhou 318000, China

²College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China

³College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China

⁴State Grid Zhejiang Electric Power Co., Ltd., Hangzhou 310007, China

Correspondence should be addressed to Wei Ruan; 2191724107@qq.com

Received 28 August 2020; Revised 19 October 2020; Accepted 15 November 2020; Published 7 December 2020

Academic Editor: Ting Yang

Copyright © 2020 Xin Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Under the background of Energy Internet, the ever-growing scale of the electric power system has brought new challenges and opportunities. Numerous categories of measurement data, as the cornerstone of communication, play a crucial role in the security and stability of the system. However, the present sampling and transmission equipment inevitably suffers from data missing, which seriously degrades the stable operation and state estimation. Therefore, in this paper, we consider the load data as an example and first develop a missing detection algorithm in terms of the absolute difference sequence (ADS) and linear correlation to detect any potential missing data. Then, based on the detected results, we put forward a missing recovery model named cascaded convolutional autoencoders (CCAEC), to recover those missing data. Innovatively, a special preprocessing method has been adopted to reshape the one-dimensional load data as a two-dimensional matrix, and hence, the image inpainting technologies can be conducted to address the problem. Also, CCAEC is designed to reconstruct the missing data grade by grade due to its priority strategy, which enhances the robustness upon extreme missing situations. The numerical results on the load data of the Belgium grid validate the promising performance and effectiveness of the proposed solutions.

1. Introduction

Nowadays, measurement data are the foundation of the power system. The massive collected data especially the quality of electricity such as voltage, current, and load are tightly associated with safe operation and economic dispatch [1]. However, due to the growing size of the power grid and the massive number of field sensors, the absence and anomalies of data measurements cannot be avoided, which is mainly due to the failures and disabilities of terminal equipment or performance degradation of transmission channels [2, 3]. The problem of missing data may lead to serious consequences including stability, optimization, and fault prevention [4]. In addition, the measured values of the missing data can be replaced by unknown noise, which makes them more difficult to be perceived and diagnosed. Thus, the efficient and accurate data detection of data anomalies and

recovery of the missing data is a fundamental for the development of data-driven analysis and advanced algorithmic solutions.

Essentially, the detection of missing data can be classified as a branch of abnormal or outlier detection, and the related researches have been investigated in the previous decades [5]. In the conventional algorithms, the methods involving residual and sudden change are discussed [6, 7]. Based on statistical analysis, 3σ criteria [8], Z-score [9], and clustering [10] are introduced. Specifically, in the power system, scholars suggest approaches combined with the correlativity of multimeasured data to improve accuracy [11]. However, these solutions are susceptible to data pollution due to the existence of missing data and lead to unsatisfactory performance.

In recent years, with the remarkable development of artificial intelligence (AI) technologies, more and more

scholars concentrate on the application of AI technologies in data recovery [12]. In literature [13], an unsupervised learning framework based on Wasserstein generative adversarial network (WGAN) is proposed to repair the missing data of active power, reactive power, voltage amplitude, and phase in power system, which achieves high accuracy, but the missing mask is required to be detected in advance, and the processing efficiency is relatively low due to the one-dimensional convolution. In [14], the adaptive neural fuzzy inference system (ANFIS) model is developed to recover the missing data of wind power. It performs better compared with traditional empirical methods but is difficult to generalize to other data. Furthermore, many other solutions have been adopted (e.g., [15–20]), but almost all the discussed solutions will deal with the missing data indiscriminately and without priority, which brings bad performance in extreme situations.

This paper will take Belgium load data (<http://www.elia.be>) as an instance to propose a new model to detect and recover the missing data. Firstly, beginning with the analysis of the characteristics of missing data, we present a detection method with ADS and linear correlation to detect the potential missing mask from the input incomplete data. Secondly, preprocessing will be applied to the incomplete data and the detected missing mask as well, which reshapes them as images (matrices). Finally, we demonstrate a CCAE model to address the missing regions in the grade of the image by grade with defined priority.

The rest of this paper will be organized as follows. In Section 2, related work and basic theories are introduced. In Section 3, the method of missing detection is investigated in Section 3.1 and then the missing recovery algorithm is developed in Sections 3.2 and 3.3. Section 4 gives some numerical results and discussion based on the load data of the Belgium grid, where a missing mask generation model is designed and employed for testing. Finally, Section 5 will conclude this paper and list the strength and weaknesses of the proposed model, and also, future work is discussed.

2. Related Work

In this chapter, the related work including abnormal detection, convolutional neural network (CNN), and autoencoder (AE) is introduced, which are the basic technologies applied in this paper.

2.1. Abnormal Detection. There many abnormal detection models widely used in the literature. The elementary idea is to model the pattern of data and then set a proper threshold or condition to pick out abnormal data in datasets. In this part, the 3σ criteria will be explored.

For the dataset that obeys the Gaussian distribution or known as a normal distribution [21], $N(\mu, \sigma^2)$, as shown in Figure 1. The mean μ and standard deviation σ can be estimated through maximum likelihood estimation (MLE). According to the features of Gaussian distribution, the

possibility of data lying in range $(\mu - 3\sigma, \mu + 3\sigma)$ is 99.7% [22]. Hence, the data out of that range could be labeled as an outlier. Even though the original data maybe do not obey the Gaussian distribution strictly but just approximately, we can adjust the 3σ properly to 2.5σ or 3.5σ for examples, which still makes it work well.

2.2. Convolutional Neural Network. A convolutional neural network (CNN) is known as a feedforward neural network with convolutional computation that is a typical image processing paradigm in deep learning [23]. CNN is capable of representation learning and can process the input image with shift-invariant classification. Therefore, it is called shift-invariant artificial neural network (SIANN) as well. An example of CNN-based classification is illustrated in Figure 2.

The convolutional computation in CNN illustrated in Figure 3 differs from that in equation (1). In CNN, the convolution is done for two-dimensional input and does not require the reverse operation to the final output:

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(\tau)g(x - \tau)d\tau. \quad (1)$$

The reason we employ CNN instead of artificial neural network (ANN) [24] to process pictures is that the parameter sharing in CNN enables us to analyze images with much fewer parameters. This is because the fully connected layers are only used in the last several layers of CNN. Hence, its training efficiency is better than ANN, when tackling with the semantics of images. Also, the training strategies can be supervised or unsupervised, which depend on the targets.

Typically, there are different categories of layers in CNN, e.g., convolutional layer, pooling layer, and fully connected layer. The convolutional layer applies the convolutional kernel to the inner product the input, region by region, and the features can be extracted in the output, as demonstrated in Figure 3.

The pooling layer is designed to reduce the output size of the convolutional layer and then diminish the required parameters in the following convolutional layers as well. Another important benefit is that the pooling layer can alleviate overfitting and increase generalization ability. The major kinds of pooling layers include max pooling and average pooling, and the computation is shown in Figure 4.

The fully connected layer is similar to the layers in ANN. The only difference is that we will first flatten the two-dimensional output of the convolutional layer or pooling layer, as a one-dimensional vector, and then, the fully connected layer is employed, as described in Figure 5. Thus, the fully connected layer is also named the flatten layer on CNN.

CNN has been widely adopted in computer vision, such as image classification [25] and object recognition [26]. And the development of autopilot is also firmly incorporated with CNN [27].

2.3. Autoencoder. Autoencoder (AE) is a kind of supervised or unsupervised ANN used for data compression,

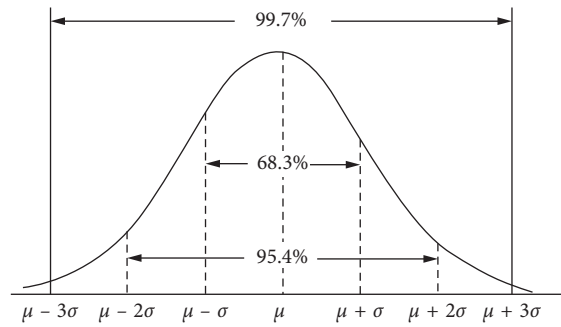


FIGURE 1: Gaussian distribution.

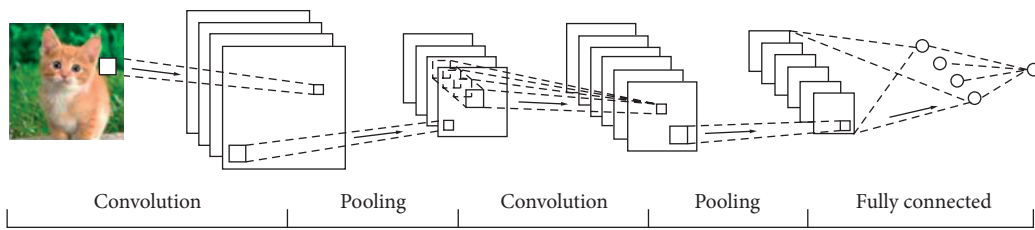


FIGURE 2: An example of CNN for classification.

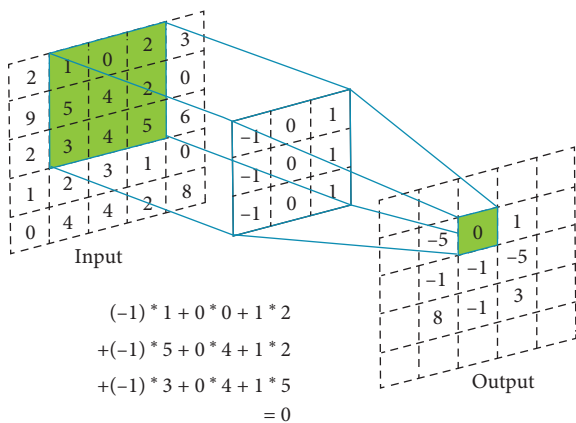


FIGURE 3: Convolutional computation on CNN.

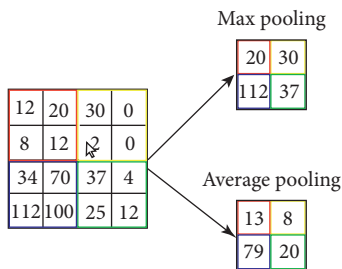


FIGURE 4: Two kinds of the pooling layer.

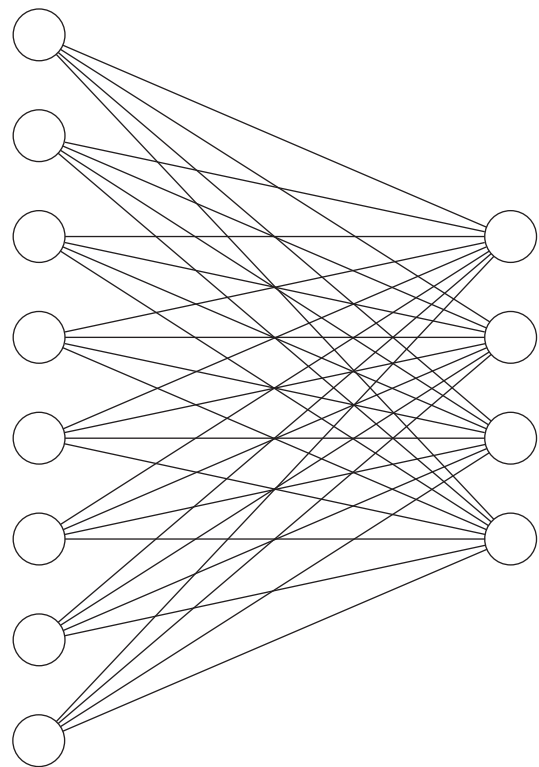


FIGURE 5: Fully connected layer.

representation learning, dimensionality reduction [28], and image denoising [29] which was first proposed by Rumelhart in 1986 [30]. An example of AE is presented in Figure 6.

The output of AE is required to be the same as the input as possible, as defined in (2) to (4). After training, AE can efficiently encode the features of input data:

$$f: X \rightarrow Z, \tag{2}$$

$$g: Z \rightarrow X, \tag{3}$$

$$f, g = \arg \min_{f, g} \|X - g[f(X)]\|^2. \tag{4}$$

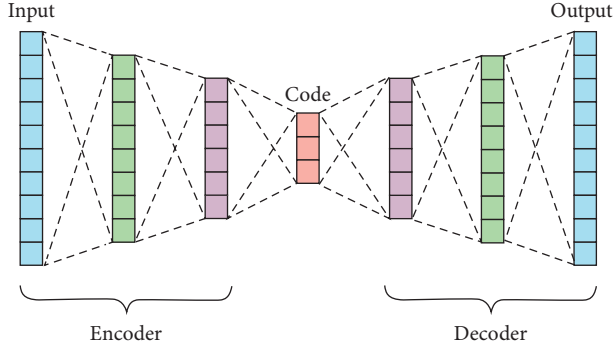


FIGURE 6: Illustration of an AE example.

As shown in Figure 6, AE has an encoder and a decoder, which essentially are fully connected layers. The encoder is responsible for feature learning, while the decoder should be able to reconstruct the input from the encoded features. Notice that when dealing with image problems, AE also can be made of convolutional layers [31] and that is exactly the basic structure of this paper.

3. Methodology

In this section, we will firstly discuss a missing detection algorithm to obtain the missing mask from the input incomplete data. Then, the input data and the detected missing mask will be further preprocessed as matrices, respectively. After that, we build a CDCAE model to recover the missing data in the matrices and reshape the matrices back as one-dimensional time series.

3.1. Missing Detection

3.1.1. Classification of Missing Segments. Usually, the missing data caused by the faults of sampling and communication equipment are mainly manifested as discrete missing points, continuous missing segments, or the combination of the two [32], as demonstrated in Figure 7. When the contact of the sampling terminal is loose or other disturbance occurs, the waveform of the data may appear as discrete missing points. However, in each link of transmission, the loss of data packet by temporary communication failure will result in a continuous missing segment.

The discrete missing points can be regarded as special cases of the continuous missing segments with a length of one. Hence, we only consider the missing segments.

In fact, the measurement values at the missing segments are usually not exactly zero or NA, but the noise is distributed near zero. This kind of noise includes not only the background noise from sampling equipment and transmission channel but also the noise due to failures or faults as well. To simplify the model, it is reasonable to assume that the noise at the missing segments is subjected to the Gaussian distribution with zero mean and relatively smaller variance than that of the normal data signal.

In this work, the missing segments are classified as typical missing segments and atypical missing segments, as illustrated

in Figures 8 and 9. Since the normal data are very likely to be far away from zero while the noise at missing segments is distributed near zero, for the beginning and the end of a missing segment, the curve will show an abnormal jump. We call those missing segments the typical missing segments, which are the most cases as well. On the contrary, there is a very low possibility that the normal data before or after a missing segment are also distributed near zero, which leads to the overlap of the ranges of the normal data and the missing data. In this situation, the curve may have an inapparent jump at the beginning or end of the missing segment. Therefore, we name those missing segments as atypical missing segments.

Here, the threshold of the abnormal jump is defined as the abnormal values in the ADS of the input sequence. The differential sequence (DS) for the load data is naturally subjected to the Gaussian distribution as well with zero mean; thus, we can simply apply the 3σ criteria to specify the abnormal values in ADS and then to locate all the abnormal jumps. However, because there might also exist abnormal jumps in atypical missing segments as indicated in Figure 9, it is still unable to locate the two kinds of missing segments.

In addition to the sudden jump of the curve, another feature might not be so noticeable. Under the small window size, the segments of normal data usually show a shape of a regular curve which has a high linear correlation for time, while the missing data segments will have a much lower linear correlation. This could be another important factor when distinguishing the missing data from normal data.

3.1.2. The Criterion of Sigma and Linear Correlation.

Based on the definition in Section 3.1.1, the detection problem can be separated as two subproblems for typical and atypical missing segments, respectively. This paper will firstly propose a method to diagnose the typical missing segments. And then with the results, the remaining atypical missing segments can be detected.

Assume the ground truth data as $X = (x_1, x_2, \dots, x_{n-1}, x_n)$, the input incomplete data as $X' = (x'_1, x'_2, \dots, x'_{n-1}, x'_n)$, the missing mask as $M = (m_1, m_2, \dots, m_{n-1}, m_n)$, and the noise signal as $W = (w_1, w_2, \dots, w_{n-1}, w_n)$, then

$$X' = X - X \odot M + W \odot M, \quad (5)$$

where m_i is a binary number and $m_i = 1$ (0) represents x_i is missing (normal), noise w_i is subjected to the Gaussian distribution $N(0, \sigma_N^2)$, and \odot is the element-wise production operator.

The steps to detect the missing mask M from the input incomplete data X' are described as follows:

- (1) Define the DS and ADS of X' as

$$\begin{aligned} \text{DS}(X') &= \{d_i | d_i = x'_i - x'_{i+1}, x'_i \in X'\}, \\ \text{ADS}(X') &= \{a_i | a_i = |x'_i - x'_{i+1}|, x'_i \in X'\}. \end{aligned} \quad (6)$$

- (2) Assume d_i is subjected to $N(0, \sigma_D^2)$ and calculate the standard deviation σ_D . Label all the a_i in $\text{ADS}(X')$

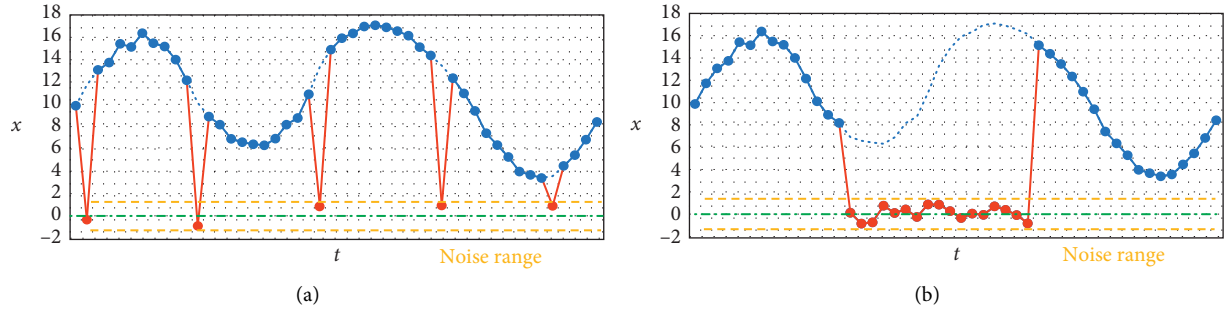


FIGURE 7: The (a) discrete and (b) continuous missing data (red) in the noise range.

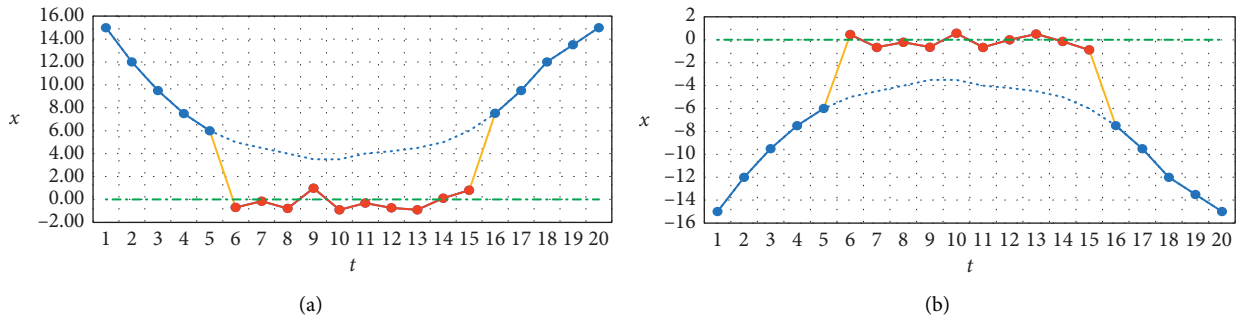


FIGURE 8: Examples of typical missing segments with two obvious jumps (yellow): (a) typical missing segment 1; (b) typical missing segment 2.

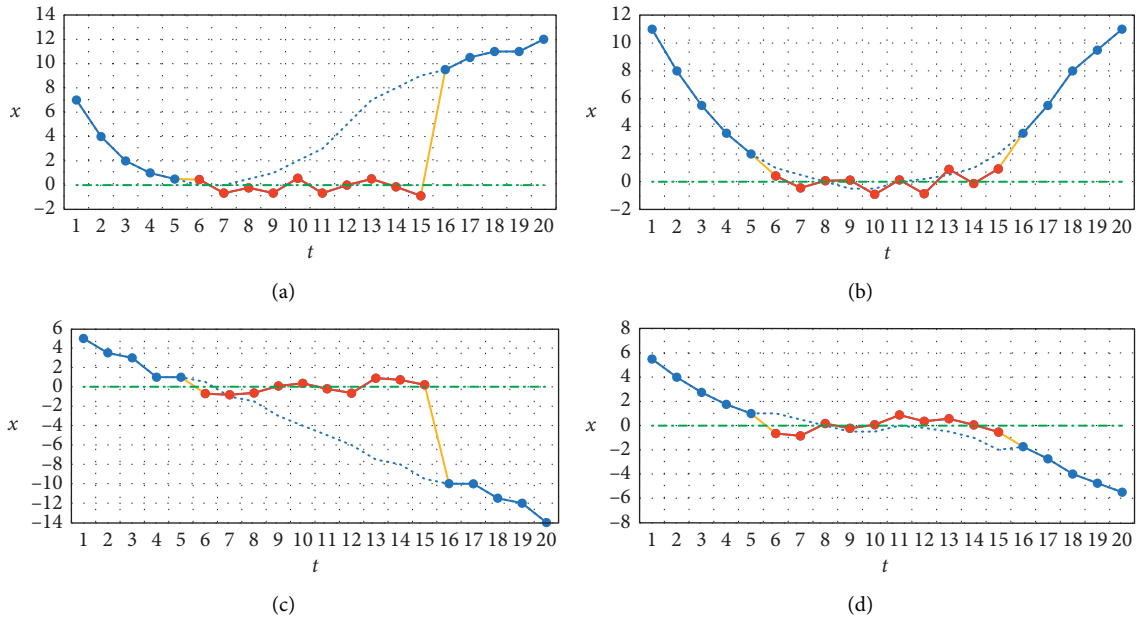


FIGURE 9: Examples of atypical missing segments with only one or none obvious jump (yellow): (a) atypical missing segment 1; (b) atypical missing segment 2; (c) atypical missing segment 3; (d) atypical missing segment 4.

- that $a_i \geq 3.5\sigma_D$ as abnormal values due to abnormal jumps.
- Compute the linear correlation r for every segment between the adjacent labeled a_i for time. If $r \leq 0.3$, the corresponding segment is labeled as a typical missing segment.

- Since the noise signal w_i in missing segments obeys $N(0, \sigma_N^2)$, we can estimate the unknown σ_N through the detected typical missing segments.
- Locate all the segments of X' within $(-2.5\sigma_N, +2.5\sigma_N)$ except the detected typical missing segments in (3), and check the linear correlation r

again. For those with $r \leq 0.3$, label the corresponding segments as atypical missing segments.

- (6) Set m_i on the detected typical and atypical missing segments as 1 and the other as 0. Obtain the detected missing mask M .

To evaluate our detection algorithm, we will refer to the confusion matrix [33] and calculate the precision P , recall R , and F_1 score [34], which is defined in 1 and in the following equations:

$$P = \frac{TP}{TP + FP}, \quad (7)$$

$$R = \frac{TP}{TP + FN}, \quad (8)$$

$$F_1 = \frac{2PR}{P + R}. \quad (9)$$

3.2. Preprocessing

3.2.1. Normalization. After the missing mask M in X' has been detected, it is necessary to normalize X' as $\hat{X}' = (\hat{x}'_1, \hat{x}'_2, \dots, \hat{x}'_{n-1}, \hat{x}'_n)$ out of convenience, and a possible choice is shown as follows [35]:

$$\hat{x}'_i = \frac{x'_i - \min(X')}{\max(X') - \min(X')}. \quad (10)$$

But due to the pollution from the Gaussian noise in the missing segments, we may not gain the real maximum and minimum values, which will lead to a gap in values distribution after normalization, and the values of data cannot fill this normalized range because the is possible even higher (lower) than the real maximum (minimum).

To avoid this problem, we will calculate the maximum and minimum values only in the normal data $\hat{X} = \{x'_i | x'_i \in X', m_i\}$ and then normalize x'_i as

$$\hat{x}'_i = (1 - m_i) \left(\frac{x'_i - \min(\hat{X})}{\max(\hat{X}) - \min(\hat{X})} 0.99 + 0.01 \right). \quad (11)$$

The values of \hat{x}'_i can well fully fill the range (0, 1), and there is no more apparent gap in the distribution of the values as well. Notice that a potential advantage by doing so is that noise in x'_i will be replaced by zero which represents missing data uniquely and vice versa. So, the following recovery algorithm can easily know which data are the missing data by just reading the zero values, even without knowing the missing mask M .

When preparing the training datasets, the ground truth data X also will be normalized as $\hat{X} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{n-1}, \hat{x}_n)$, where

$$\hat{x}_i = \frac{x_i - \min(X)}{\max(X) - \min(X)} 0.99 + 0.01. \quad (12)$$

It should be noted that the local maximum and minimum values used here for normalization may not be the

TABLE 1: Confusion matrix in detection.

| | Missing | Normal |
|-------------------|---------------------|---------------------|
| Predicted missing | True positive (TP) | False positive (FP) |
| Predicted normal | False negative (FN) | True negative (TN) |

global maximum and minimum of the normal data before missing happens, for the reason the real maximum and minimum could be blocked by the missing mask. Thus, when we normalize those blocked data, the results are probably beyond range (0, 1). But fortunately, since the data discussed in this paper are load data with obvious periodicity, the local maximum and minimum would be very close to the global maximum and minimum.

3.2.2. Grade. Repairing the missing data is to figure out how to make the best estimation for the missing data based on the adjacent normal data. Hence, making full use of the adjacent normal data is the key to solve the problem.

We note that, for different missing data at the same missing segment, the specific locations of the missing data are different, and also the numbers of adjacent available normal data are different. In detail, the missing data near the beginning or the end of the missing segments are closer to the normal data, which makes them easier to be recovered, while the missing data at the center of the missing segments are far away from the normal data and difficult to be addressed.

To design more targeted recovery algorithms for different missing situations, this paper will introduce powerful improvement on the detected missing mask M in Section 3.1.2. The core idea is that the missing data at the center should be recovered based on the recovery of the missing data at edges, which indicates the edge missing data have a higher priority and are before being recovered:

$$T = \text{Grade}(M). \quad (13)$$

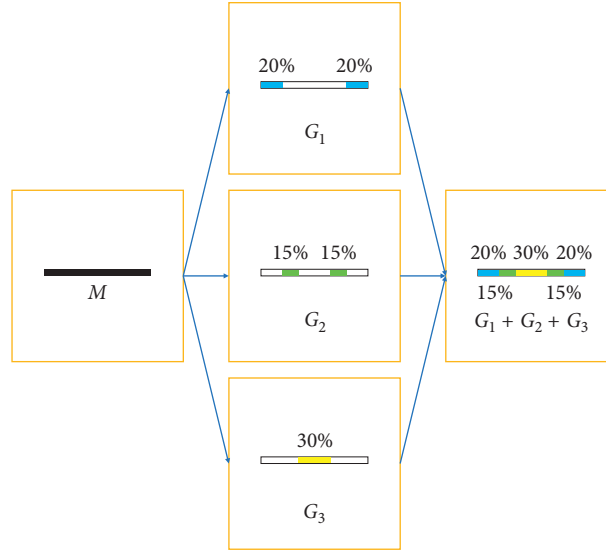
To distinguish the missing data at different positions, we grade M into K submissing masks $T = (G_1, G_2, \dots, G_{K-1}, G_K)$ separately, where K is the grade parameter and the j -th submissing mask is $G_j = (g_1^{(j)}, g_2^{(j)}, \dots, g_{n-1}^{(j)}, g_n^{(j)})$. The missing segments in M will be divided into smaller submissing segments from two ends toward the inside. The ratio of the submissing segments in G_j concerning that in M is defined as R_j :

$$\sum_{j=1}^K G_j = M, \quad (14)$$

$$\sum_{j=1}^K g_i^{(j)} = m_i,$$

$$\sum_{j=1}^K R_j = 1,$$

where $g_i^{(j)}$ is a binary number similar as m_i .


 FIGURE 10: Submitting masks G_1 , G_2 , and G_3 from the missing mask M .

In this paper, the hyperparameters $K = 3$ and corresponding $R_1 = 40\%$, $R_2 = 30\%$, and $R_3 = 30\%$, as shown in Figure 10.

3.2.3. Reshape. The load data in the power system change along with the patterns of society operation and production. Hence, it will show evident multiple periodicities in days, weeks, quarters, and years. When repairing this kind of data, only referring to the continuity between the directly adjacent normal data and the missing data (e.g., data before and after half an hour) is not enough and leads to bad performance. Instead, the periodicity should be taken into consideration, meaning that we have to also refer to the data in adjacent cycles (e.g., the data at the same time but different periods), since those data, to some extent, are indirectly adjacent and share very similar patterns, as presented in Figure 11.

In terms of semantic intensity, direct adjacency is stronger than indirect adjacency, but they can be combined as a reference. For the data at the edges of the missing segments, due to the constraint of the waveform continuity, the directly adjacent data are nearly the most important repair reference; but for the missing data inside the missing segment, there will not be any directly adjacent data for reference, while the indirect adjacent data become a very significant factor instead. However, traditional mathematical estimation and interpolation methods can only perceive data inside a very limited window around the missing data, so they cannot make full use of the information from indirect adjacent data. As a result, the repair accuracy is low, especially for the long continuous missing segments.

When dealing with similar problems, literature [35] proposes a method to transfer the one-dimensional harmonic data into a two-dimensional grayscale image by

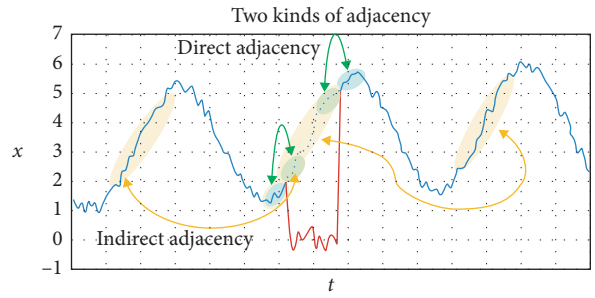


FIGURE 11: Direct and indirect adjacency relationship in load data.

periodic truncation and reshaping, as demonstrated in Figure 12 $n = km$. Inspired by this, this paper will reshape the one-dimensional incomplete data \widehat{X}^l and corresponding submitting masks G_j into matrices.

Take \widehat{X}^l , for example, assume the number of sampling points per day is m , for a dataset with k days, and the size . Then, reshape \widehat{X}^l into a k -by- m matrix $A_{\widehat{X}^l}$:

$$A_{\widehat{X}^l} = \text{reshape}(\widehat{X}^l) = [a_{i,j}]_{k \times m}, \quad (15)$$

where $a_{i,j} = \widehat{x}'_{(i-1)m+j}$. For the load data of Belgium grid, $k = 2000$ and $m = 96$.

And similarly, matrices A_{G_j} can be obtained. The two-dimensional structure of the matrix enables the direct and indirect adjacency to be compatible with each other in rows and columns separately. And the shaped matrix can be understood as a special “generalized” image. Based on the above analysis, when we reshape the data, the problem of repairing the missing one-dimensional data becomes the problem of inpainting a two-dimensional image. In

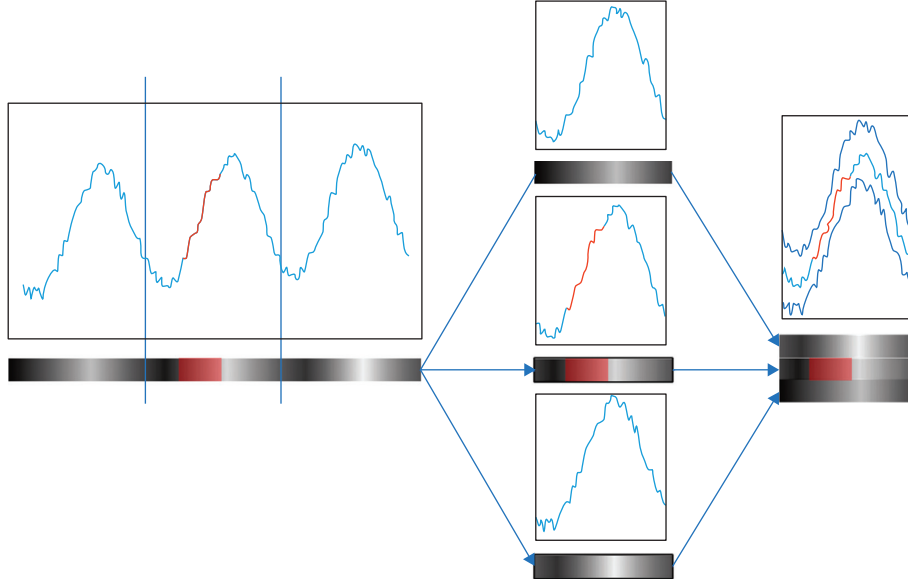


FIGURE 12: Reshaping of the one-dimensional data to the two-dimensional image.

the following discussion, we will combine the deep learning and image processing techniques to address this problem.

3.2.4. Edge Padding. After reshaping, the data located in the center of the matrices will have more available adjacent data than that at the edge of the matrices, which indicates the number of available adjacent data is radially attenuated outward from the core of the matrices, making the recovery of edges data more difficult than that of the central data.

To solve the problem of an unbalanced distribution of the available adjacent data in the radial direction, the most direct and effective method is to arrange some additional data on the edges. When we truncate the original one-dimensional data and reshape it into matrices, two adjacent data at the truncated point will be separated and then arranged at the end of this row and the beginning of the next row accordingly. So, the left edge and the right edge of the matrix are - "adjacent- " but misplaced by one row, as illustrated in Figure 13. Thus, the data in the left and right edges can be used as padding data for each other.

Take $A_{X'}^{\wedge}$, for example, to improve this unbalanced distribution, two k -by- p padding matrices $B_{X'}^{\wedge}$ and $C_{X'}^{\wedge}$ are designed, and p is the padding depth:

$$B_{X'}^{\wedge} = [b_{i,j}]_{k \times p}, C_{X'}^{\wedge} = [c_{i,j}]_{k \times p}, \quad (16)$$

where

$$b_{i,j} = \begin{cases} a_{i-1,m-p+j} & i > 1 \\ 0 & i = 1 \end{cases}, \quad (17)$$

$$c_{i,j} = \begin{cases} a_{i+1,j} & i < k \\ 0 & i = k \end{cases}$$

Define the ratio of p and m as hyperparameter padding ratio η :

$$\eta = \frac{p}{m}. \quad (18)$$

Then, arrange $B_{X'}^{\wedge}$ and $C_{X'}^{\wedge}$ to the left and right sides of $A_{X'}^{\wedge}$, respectively, to get a k -by- L padding matrix $Z_{X'}^{\wedge}$ as

$$Z_{X'}^{\wedge} = \text{Padding}(A_{X'}^{\wedge}) = [B_{X'}^{\wedge} A_{X'}^{\wedge} C_{X'}^{\wedge}] = [z_{i,j}]_{k \times L}, \quad (19)$$

where $L = m + 2p$.

And similarly, padding matrix Z_G can be attained.

3.2.5. Slice. Because there are no padding data on the upper and lower sides of padding matrices, we will cut the padding matrices into smaller L -by- L slices and set proper overlapped rows as padding data on the upper and lower sides.

Take $Z_{X'}^{\wedge}$, for example, the slices are defined as $S_{X'}^{\wedge}$, where the t -th slice $S_{X'}^{\wedge}(t)$ is a L -by- L matrix:

$$S_{X'}^{\wedge} = \text{Slice}(Z_{X'}^{\wedge}), \quad (20)$$

$$S_{X'}^{\wedge}(t) = [s_{i,j}^{(t)}]_{L \times L},$$

where $s_{i,j}^{(t)} = z_{(t-1)m+i,j}$ and $t \leq n_s = \lfloor k - 2p/m \rfloor$.

For adjacent slices $S_{X'}^{\wedge}(t)$ and $S_{X'}^{\wedge}(t+1)$, there are $2p$ overlapped rows on the lower side of $S_{X'}^{\wedge}(t)$ and the upper side of $S_{X'}^{\wedge}(t+1)$. Hence, the first and last p rows and columns are redundant data. And as a result, when recovering a slice, we only need to consider its center m -by- m region which is defined as the core region:

$$\text{Core}(S_{X'}^{\wedge}(t)) = [u_{i,j}^{(t)}]_{m \times m}, \quad (21)$$

where $u_{i,j}^{(t)} = s_{i+p,j+p}^{(t)}$.

In particular, the core areas of the first (last) slice will include the upper (lower) p rows, which leads to a $(m+p)$ -by- m matrix:

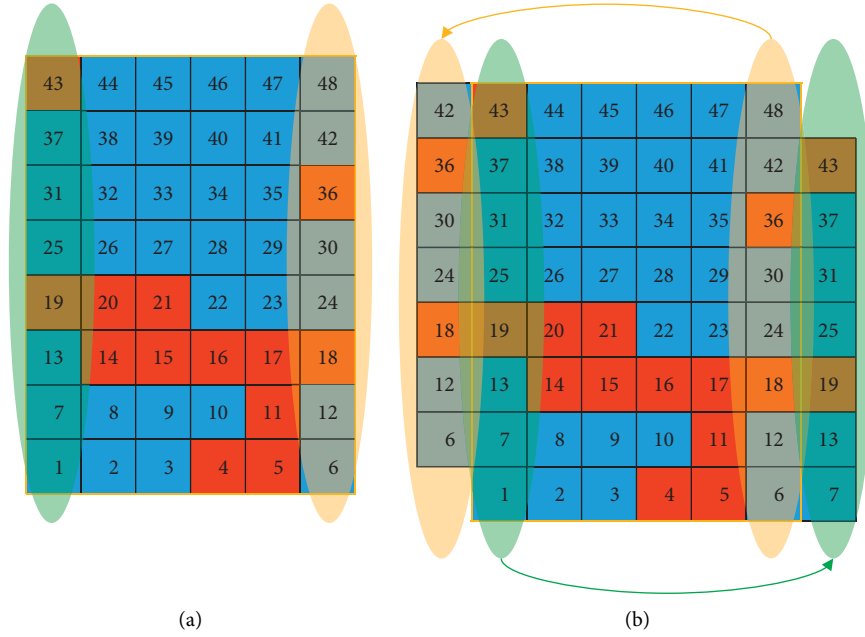


FIGURE 13: Misplacement adjacency between the left and right sides and self-padding.

$$\begin{aligned} \text{Core}\left(S_{X'}^{\wedge}(1)\right) &= \left[u_{i,j}^{(1)}\right]_{(m+p) \times m}, \\ \text{Core}\left(S_{X'}^{\wedge}(n_s)\right) &= \left[u_{i,j}^{(n_s)}\right]_{(m+p) \times m}, \end{aligned} \quad (22)$$

where $u_{i,j}^{(1)} = s_{i,j+p}^{(1)}$ and $u_{i,j}^{(n_s)} = s_{i+p,j+p}^{(n_s)}$.

Similarly, slices S_{G_j} can be obtained.

3.3. Missing Recovery. When we use a two-dimensional matrix to represent one-dimensional data, the problem of recovering one-dimensional data becomes the problem of image inpainting. Recently, CNN and GAN technologies in the deep learning field have excellent performance on image inpainting. In [36], the author uses the previous five convolutional layers from the AlexNet [23] as an encoder to extract the features of images with missing areas and then uses six deconvolutional layers as a decoder to restore the missing regions from the learned features. Inspired by this, this paper will put forward a convolutional autoencoder-based network to recover the missing data in the pre-processed matrices.

As we know, the feature learning in the AlexNet is designed for the object classification problem, where the object positions will not matter since the category of an object does not depend on its position. And because of the use of the max pool, “valid” padding, and flatten layer, the size of the tensor will be compressed, which will blur the position information. Therefore, it is nearly impossible to trace back to the original positions of features in deep layers, especially when the input data are damaged heavily.

For reshaped matrices, because the semantic information is not uniformly distributed in rows and columns, the

original position of the feature is even more important when extracting the features. If the convolution and deconvolution framework is applied directly, the fuzzy location of features in deep layers will further degrade the situations. Moreover, since the “generalized” images are usually low-dimensional with low rank, the context encoder in [36] will have bad performance as the author reminded.

Thus, this paper presents a CCAE network based on the context encoder as follows: only use convolutional layers without any flatten, pooling, or fully connected layers, replace the padding mode from “valid” to “same,” and set the stride as one which keeps the height and width of the output tensors in every layer equal to the input, namely, L -by- L . Finally, the output matrix will be restored to one-dimensional.

3.3.1. Network Structure. To recover the preprocessed incomplete data $S_{X'}^{\wedge}$ with S_{G_j} , a CCAE model is proposed with the structure shown in Figure 14.

There are K convolutional autoencoders (CAE_j) blocks cascaded in CCAE corresponding to K submitting masks S_{G_j} . Each CAE_j has an encoder E_j , a decoder D_j , and a filter F_j . Encoder E_j and decoder D_j are made of Q convolutional layers, respectively, where the stride is one, padding mode is “same,” and activation function is Relu. The filter F_j is used to update the recovery results of the missing segments in S_{G_j} , that is,

$$F_j = F_{j-1} + D_j \odot S_{G_j}. \quad (23)$$

And then, F_j will be the input of E_{j+1} in CAE_{j+1} , which ensures $S_{G_{j+1}}$ will be recovered based on the recovery result of S_{G_j} . In particular,

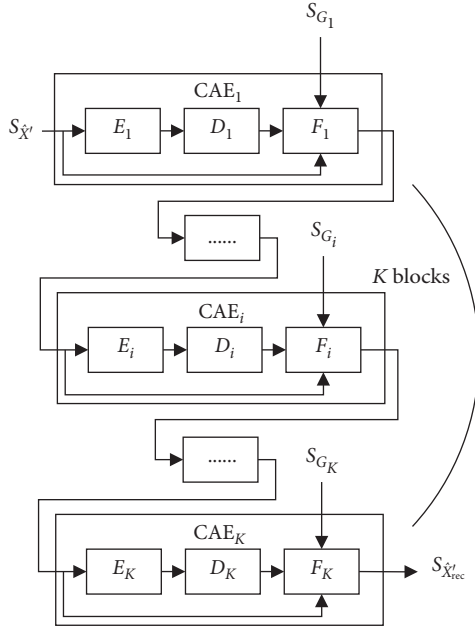


FIGURE 14: General structure of CCAE.

$$\begin{aligned} F_1 &= S_{X'}^{\wedge} + D_1 \odot S_{G_1}, \\ S_{X'_{rec}}^{\wedge} &= F_K, \end{aligned} \quad (24)$$

where $S_{X'_{rec}}^{\wedge}$ is the recovery result of $S_{X'}^{\wedge}$.

In this paper, $K = 3$ as mentioned and $Q = 2$. As a result, there are four convolutional layers in each CAE_j . The number of convolutional kernels in each layer is 64, 96, 32, and 1 in CAE_1 , 32, 64, 16, and 1 in CAE_2 , and 32, 64, 16, and 1 in CAE_3 . The corresponding kernel size is (5, 5), (11, 11), (5, 5), and (3, 3) in CAE_1 ; (7, 7), (5, 5), (3, 3), and (3, 3) in CAE_2 ; and (5, 5), (5, 5), (3, 3), and (3, 3) in CAE_3 , as demonstrated in Figure 15. The red rectangles represent the core areas in Section 3.2.5.

Without a flatten layer and fully connected layers, we only employ two convolutional layers for feature learning. As a result, the learned feature will locate at its original position, which means this network will encode a feature vector for every point in the input matrix and just put that feature vector at the same position as the computed point. Through that, the position information can be retained during feature learning to the most extend.

3.3.2. Loss Function. One thing we should notice is that the size of the output matrices of CCAE is still L -by- L , in which only their m -by- m core areas matter as mentioned in Section 3.2.5. Hence, the loss function \mathcal{L} is defined as the root mean squared error (RMSE) of the missing data within core areas:

$$\mathcal{L}(S_{X'_{rec}}^{\wedge}, S_{X'}^{\wedge}) = \sqrt{\frac{1}{\sum_{i=1}^n m_i} \sum_{t=1}^{n_s} \left\| \text{Core}(S_{X'_{rec}}^{\wedge}(t) - S_{X'}^{\wedge}(t)) \right\|_2^2}, \quad (25)$$

where $S_{X'}^{\wedge}$ is obtained through the same preprocessing in Section 3.2.

3.3.3. Restore. Since $S_{X'_{rec}}^{\wedge}$ is a set of normalized L -by- L matrix slices $S_{X'_{rec}}^{\wedge}(t)$, it should be restored back to one-dimensional time series, which means the reverse operations of preprocessing in Section 3.2. Assume $n_s = \lfloor k - 2p/m \rfloor = k - 2p/m$, then reorganize the core areas of slices $S_{X'_{rec}}^{\wedge}(t)$ as a k -by- m matrix A_F :

$$A_F = \begin{bmatrix} \text{Core}(S_{X'_{rec}}^{\wedge}(1)) \\ \vdots \\ \text{Core}(S_{X'_{rec}}^{\wedge}(n_s)) \end{bmatrix} = [f_{i,j}]_{k \times m}. \quad (26)$$

After that, reshape A_F into a n -by-1 time series $\hat{Y}_{rec} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{n-1}, \hat{y}_n)$, where $\hat{y}_h = f_{i,j}$ for those $(i-1)m + j = h$. Finally, reverse the normalization in Section 3.2.1 to get the restored results $Y_{rec} = (y_1, y_2, \dots, y_{n-1}, y_n)$ in which

$$y_h = (\hat{y}_h - 0.01) \frac{\max(\hat{X}) - \min(\hat{X})}{0.99} + \min(\hat{X}). \quad (27)$$

4. Experimental Results and Discussion

In this section, we will validate our detection and recovery algorithms. Load data from the Belgium grid will be conducted as training and test data, and a missing mask generation model is presented to produce generative missing masks under different parameters.

4.1. Experimental Design

4.1.1. Missing Mask Generation Model. In the original load data of the Belgium grid, there are very few missing segments; therefore, it could be regarded as ground truth data X . Then, we have to manually generate missing masks M for detection and recovery testing.

Define the missing rate as γ where

$$\gamma = \frac{\sum_{i=1}^n m_i}{n}. \quad (28)$$

If we just randomly select some segments in X as missing segments, the segments collision might happen, as demonstrated in Figure 16, which results in a lower missing rate than the given γ . Therefore, a stratified sampling model is proposed to solve the problem, as shown in Figure 17.

Define the number of missing data as NMD, namely,

$$\text{NMD} = \sum_{i=1}^n m_i = n\gamma. \quad (29)$$

Then, divide NMD as NMS segments where NMS is the number of missing segments and

$$\text{NMS} = \text{random}([NMD\alpha], [NMD\beta]), \quad (30)$$

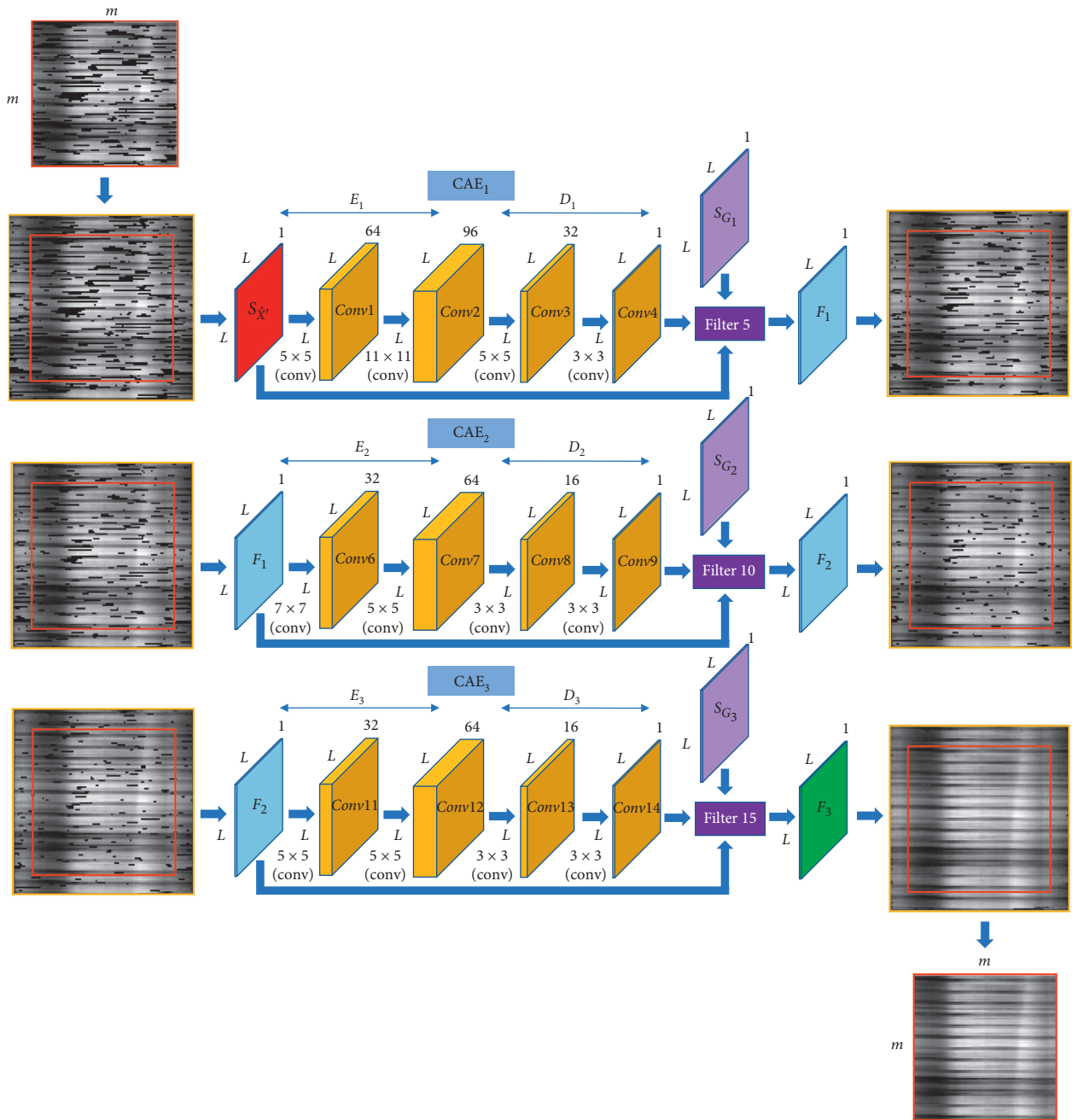


FIGURE 15: An instance of CCAE with $K = 3$ and $Q = 2$.

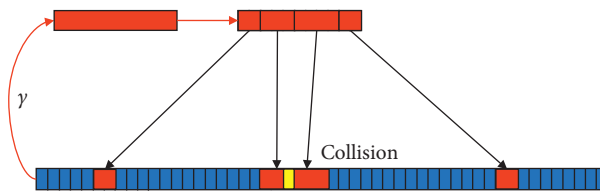


FIGURE 16: Segments collision in random generation of the missing mask.

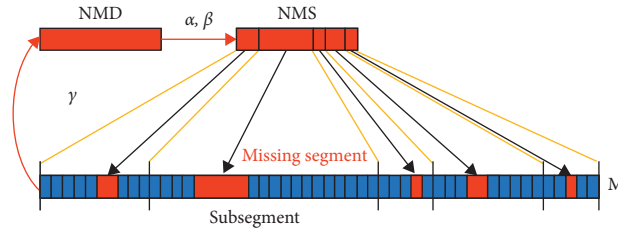
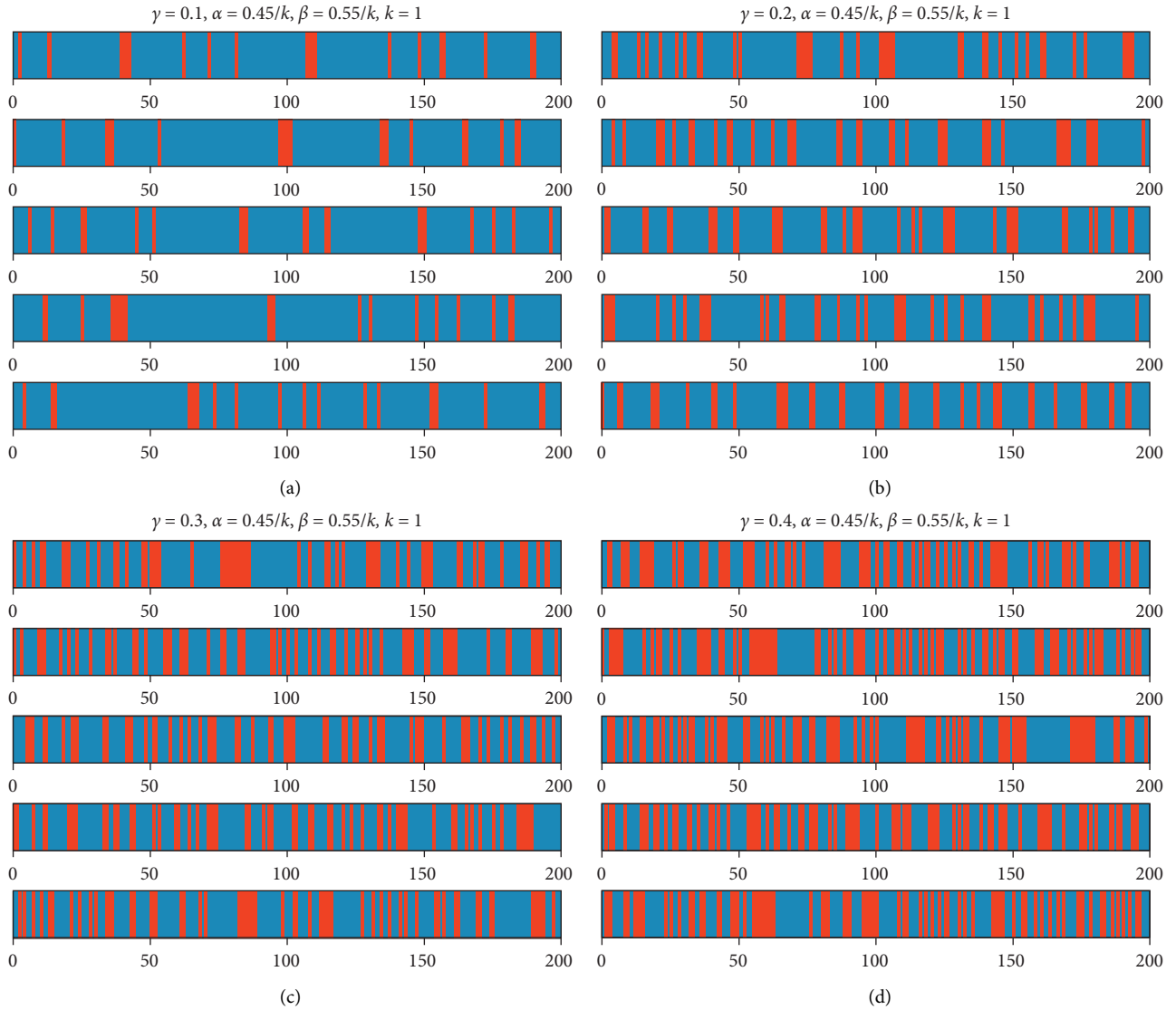


FIGURE 17: Stratified sampling generation model of the missing mask.

FIGURE 18: The missing masks in different missing rate γ .

where α and β are length parameters of missing segments which determine the average length of missing segments ALMS:

$$\text{ALMS} = \frac{2}{\alpha + \beta}. \quad (31)$$

Randomly generate an integer between $\text{NMD}\alpha$ and $\text{NMD}\beta$ as NMS and then stochastically divide NMD as NMS segments; then, according to the proportion of each NMS segment, divide M as NMS subsegments, too. Finally, within every subsegment in M , we independently select a missing segment and set those bits inside the subsegments as 1 and

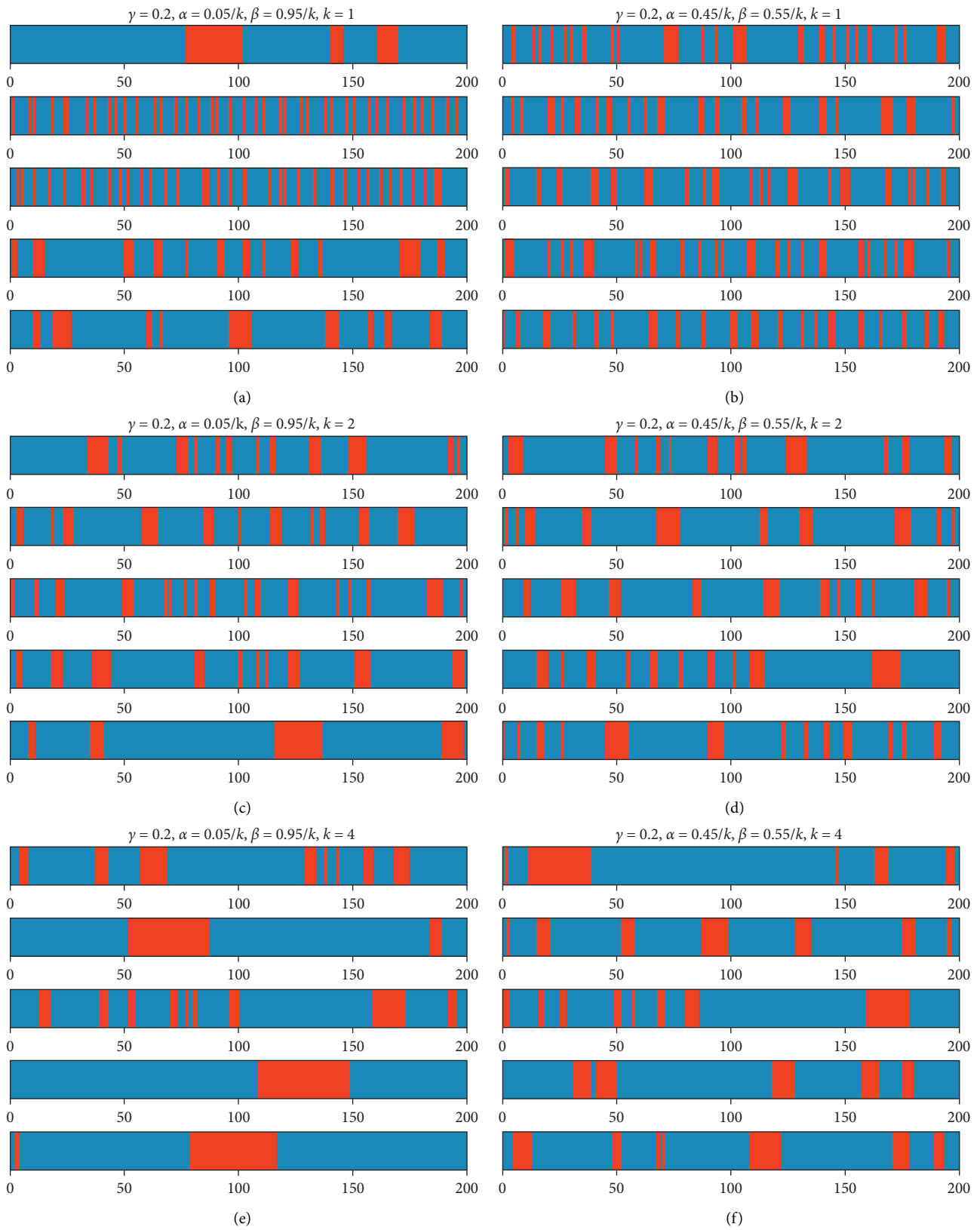


FIGURE 19: The missing masks in different α and β (ALMS).

TABLE 2: Dataset configurations for missing detection.

| # | Range | γ | SNR |
|---|-------------|----------------|----------------|
| 1 | (-500, 0) | {5%, 10%, 20%} | (15 dB, 40 dB) |
| 2 | (-250, 250) | {5%, 10%, 20%} | (15 dB, 40 dB) |
| 3 | (0, 500) | {5%, 10%, 20%} | (15 dB, 40 dB) |

TABLE 3: Dataset configurations for missing recovery.

| # | p | γ | (α, β) |
|---|-----|----------------|--|
| 1 | 0 | {5%, 10%, 20%} | {{(0.1, 0.15), (0.05, 0.075), (0.025, 0.0375)} |
| 2 | 7 | {5%, 10%, 20%} | {{(0.1, 0.15), (0.05, 0.075), (0.025, 0.0375)} |
| 3 | 9 | {5%, 10%, 20%} | {{(0.1, 0.15), (0.05, 0.075), (0.025, 0.0375)} |
| 4 | 11 | {5%, 10%, 20%} | {{(0.1, 0.15), (0.05, 0.075), (0.025, 0.0375)} |

others as 0. This pipeline ensures γ and the distribution of missing segments are independent.

Some of the generative missing masks with given γ , α , and β are shown in Figures 18 and 19. For each set of parameters, four missing masks are generated independently, and the total number of data is $n = 200$; the blue regions represent normal data with $m_i = 0$, while red regions mean missing data with $m_i = 1$.

4.1.2. Dataset Configurations. As mentioned, we will take the load data of the Belgium grid during 2014–2020 as an example, which involves 2000 days in total and 96 sampling points per day. The original data values range from 7000 MW to 14000 MW.

To better evaluate the proposed model, the detection and recovery components will be tested independently, which means the missing mask for the recovery component is the generative missing mask instead of the detected missing mask by the detection component. Hence, different datasets configuration will be used for the two components.

For the detection component, we assume noise W obeys the Gaussian distribution $N(0, \sigma_N^2)$ and define SNR for the normal data signal power P_{data} and noise signal power P_{noise} as

$$\text{SNR} = 10 \lg \frac{P_{\text{data}}}{P_{\text{noise}}}, \quad (32)$$

where $P_{\text{data}} = 1/n \|X\|_2^2$ and $P_{\text{noise}} = \sigma_N^2$.

In the experiment, we consider SNR ranges from 15 dB to 40 dB. Since the original data range from about 7000 to 14000, if we directly generate a missing mask on that, nearly all the missing segments will be typical missing segments. To comprehensively evaluate the missing detection algorithm, we linearly map the original data values into (-500, 0), (-250, 250), and (0, 500), respectively. The factor of missing rate γ should be investigated as well, which is set as 10%, 20%, and 40% separately. Besides, length parameters are fixed as $\alpha = 0.1$ and $\beta = 0.15$, then ALMS = 8. The configurations are shown in Table 2. The indexes to assess the detection results are precision P , recall R , and F_1 score in Section 3.1.2.

TABLE 4: Specifications of the software.

| Item | Version |
|------------|---------|
| Python | 3.7.6 |
| TensorFlow | 1.14 |

TABLE 5: Specifications of hardware.

| Item | Specifications |
|--------|--------------------------------|
| System | Ubuntu 19.04 |
| CPU | i9-9820x |
| GPU | Nvidia Titan Xp 12G \times 4 |
| RAM | 64G |

For the recovery component, because the noise will be cleared as zero in Section 3.1.2, we do not care about the influence of noise W . Instead, parameters γ , α , β , and p are studied. γ will be set as 5%, 10%, and 20%, while α and β will be set as (0.1, 0.15), (0.05, 0.075), and (0.025, 0.0375), corresponding to ALMS of 8, 16, and 32. Furthermore, padding depth p will be set as 0, 7, 9, and 11. For each p , the missing mask contains mixed 3×3 combinations of (γ, α, β) . The configurations are listed in Table 3. The index to assess the recovery results is RMSE in Section 3.3.2.

In addition, the missing masks with the above configurations will be generated for ten times independently to avoid the stochastic disturbance in results.

4.1.3. Training Settings. For each set of configurations in Table 3, 80% will be used as training sets and the remaining 20% will be testing sets. The specifications of software and hardware are presented in Tables 4 and 5. The optimizer is ‘‘Adam,’’ the learning rate is set to descend exponentially along the epochs, and the batch size is 20 (slices of $S_{X'}^{\wedge}(t)$ with corresponding missing masks).

4.2. Results and Discussion of Missing Detection. As illustrated in Figure 20, for the data mapped in the positive range (0, 500) or the negative range (-500, 0), all the precision, recall, and F_1 are all nearly 100% when the SNR is over 20 dB, while the SNR needs to be more than 30 dB to reach the same result for the data mapped in (-250, 250). This is



FIGURE 20: Missing detection results for different missing rates and normalization ranges.

because the ratio of the atypical missing segments for data mapped in $(-250, 250)$ is much higher than the other two, and the noise on the missing segments in that situation will have a very high possibility to be overlapped with the normal data. Moreover, we might be unable to obtain enough typical missing segments for noise estimation, which influences the detection performance.

When the SNR decreases from 20 dB to 15 dB, there is significant deterioration in the results. F_1 of data mapped in $(-500, 0)$ and $(0, 500)$ can drop to 0.7. And it will be even worse for the data mapped in $(-250, 250)$, where F_1 can be lower than 0.6. But fortunately, in most cases, the data are

TABLE 6: RMSE of CCAE for different padding depths.

| # | p | CCAЕ ($\times 10^{-2}$) |
|---|-----|---------------------------|
| 1 | 0 | 4.15 |
| 2 | 7 | 3.94 |
| 3 | 9 | 3.79 |
| 4 | 11 | 3.84 |

always positive or negative, which is far away from zero, and the noise on missing data is slight enough which ensures high SNR. Thus, the detected mask could be regarded as the ground truth missing masks. This is also

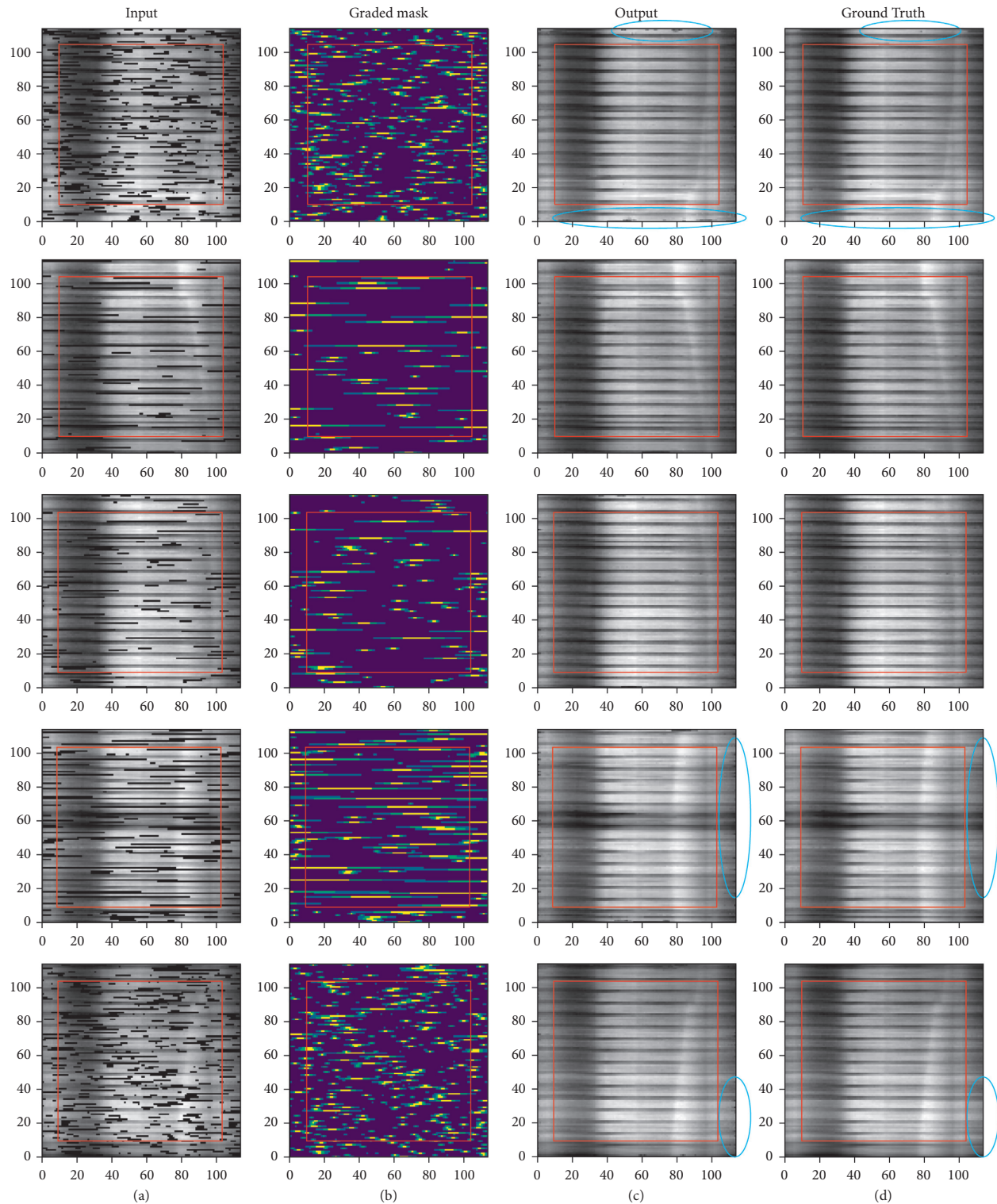
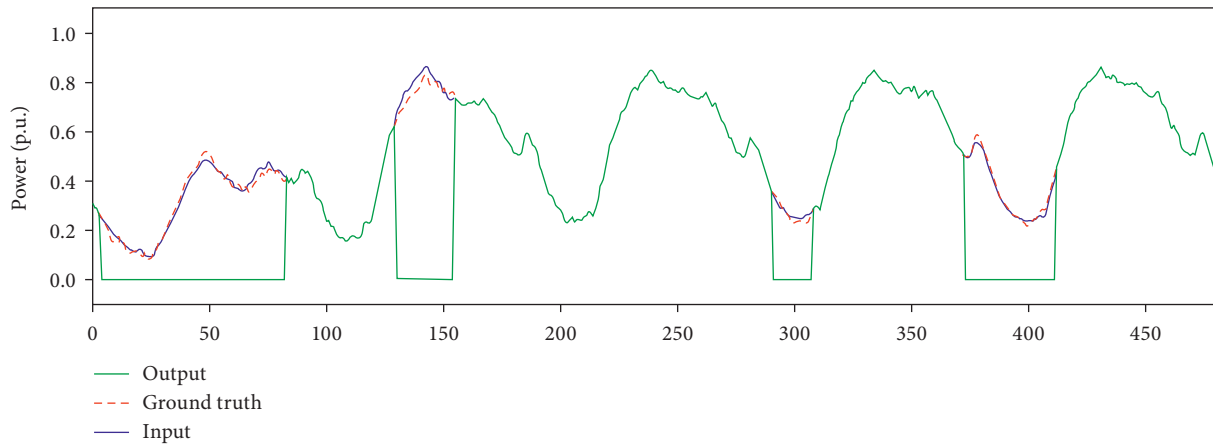


FIGURE 21: Missing recovery results with core areas inside red rectangles.

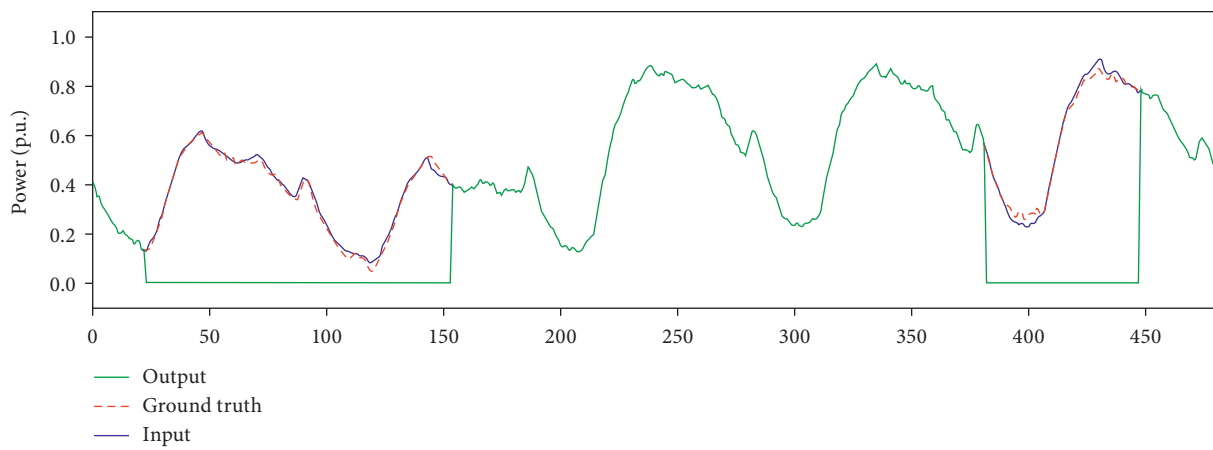
the reason for just using the generative missing mask rather than the detected missing mask to test the sequential recovery component.

Another phenomenon is that, in the high SNR region, the missing rate seemingly makes no difference to the detection, while in the low SNR region, the higher the

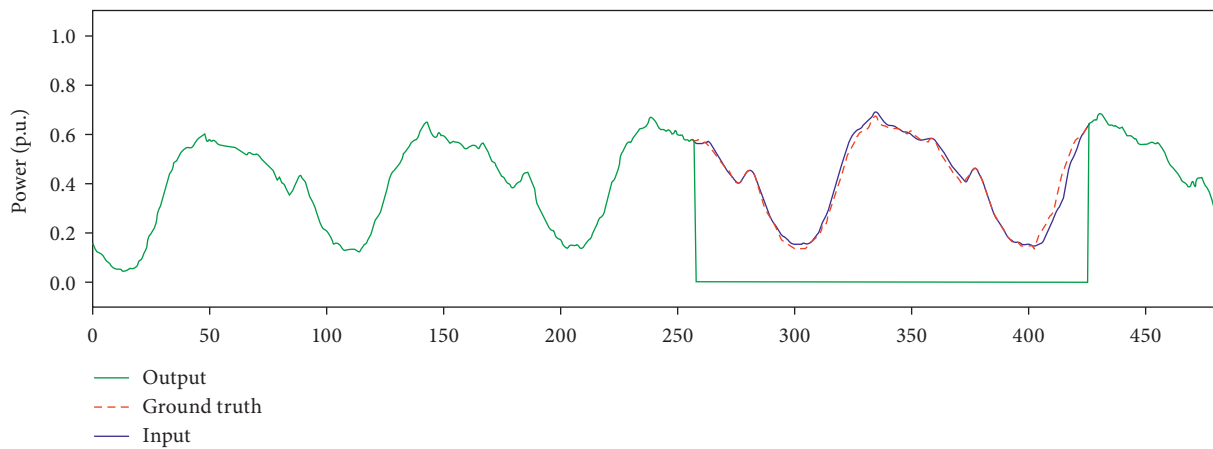
missing rate is, the better the detection performs. It may be not so intuitive. But our further analysis indicates that when the SNR is low, the ratio of the false-positive samples will increase greatly because of the overlap of normal data and noise, even when the missing rate is zero. Therefore, a higher missing rate will bring more positive



(a)



(b)



(c)

FIGURE 22: Continued.

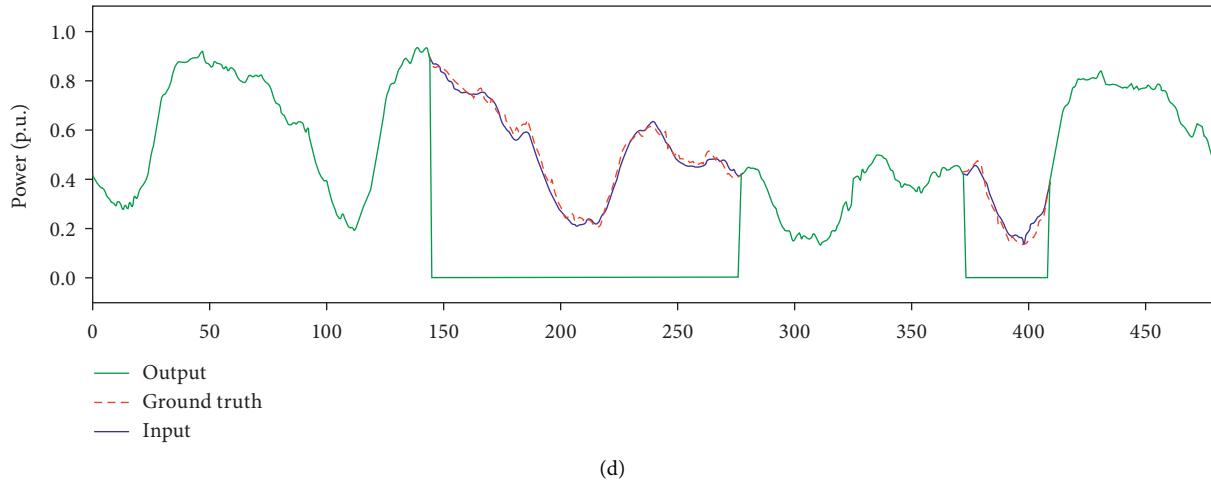


FIGURE 22: Missing recovery results in one-dimensional (96 points/day).

samples in turn, which decreases the false-positive samples to some degree.

4.3. Results and Discussion of Missing Recovery. The RMSE for different padding depths is shown in Table 6. Generally, the CCAE model performs well on the missing recovery problem, and RMSE is pretty low for the normalized range (0, 1). Besides, the edge padding technique can further improve the recovery error, but too deep padding depth may lead to a drop in the performance, and in this experiment, the optimal choice is 9 with the padding ratio of 9.375%.

During training, CCAE is allowed to output high error on those padding areas, which makes the loss function converge more easily and efficiently. And without the padding areas, those errors would appear in the core areas unavoidably, which is also the reason we design the edge padding technique.

Theoretically, the deeper the padding depth is, the more the adjacent data can be used for the edges data in the core area, but with the increase in the depth, the distance between the padding data and core data will grow linearly, too. And once beyond a certain distance, the padding data can no longer provide any useful information, which instead causes a lower proportion of the core area in the padding slice and brings low computation efficiency.

To demonstrate the recovery performance of CCAE with $p = 9$, we randomly chose some output slices of $S_{X_{rec}}^{\wedge}$ compared with the input slices $S_{X'}^{\wedge}$ and the ground truth S_X^{\wedge} in Figure 21, where those matrices are virtualized by grayscale images. The size of each slice is 114-by-114, and the blue, green, and yellow segments in submitting masks represent the submitting segments in G_1 , G_2 , and G_3 , respectively. The areas inside red rectangles are core areas.

The core areas of the output images are nearly the same as core areas of the ground truth, even for those inputs with high missing rate and long missing segments. We even cannot tell the difference between the cores of output and ground truth. Only when zooming up the output, we can

find some slight difference in the textures for the ground truth.

While in the edge padding area outside the core areas, there are obvious black holes as the blue circles shown in Figure 21 because of ignoring those areas when defining the loss function in Section 3.3.2. In Figure 22, we restore some rows of the output to one-dimensional to get \hat{Y}_{rec} , and the results are consistent with the previous discussion.

5. Conclusion and Future Research

This paper proposes a missing load data detection and recovery model based on CCAE. In the detection issue, we combine ADS and the linear correlation as a criterion to detect the potential missing segments. And based on the detection results, we further divide the detected missing mask into submitting masks with priority and then reshape the original one-dimensional data and mask into two-dimensional matrices for data enhancement. The constructed matrices are regarded as “generalized” images, which transform the recovery problem to images inpainting. Furthermore, the deep learning technologies are conducted, and we have designed a CCAE model to repair the input damaged matrices. To assess the algorithms, we build a missing mask generation model to generate missing masks. Numerical results on the load data of the Belgium grid indicate that the developed detection and recovery algorithms have satisfactory performance under different missing situations. It should be highlighted that the proposed intelligent detection and recovery solution can be used for other forms of time-series dataset.

Here, the strength of the proposed detection and recovery algorithms can be summarized as follows: it can be found that the missing detection is nearly 100% accurate for most situations; the missing segments can be recovered grade by grade with priority in submitting masks strategy, which ensures the recovery accuracy even for long-missing segments. Also, the reshaping from one-dimensional time series to the two-dimensional image is a powerful data enhancement method for the load data, which enables the

CNN to understand the semantics of one-dimensional data. Finally, the structure of CCAE is not sensitive to the input size, so it is easy to make transfer learning to datasets with different periods.

On the contrary, the proposed solution is still needed for further investigation as it has the following potential limitations: firstly, under the condition of low SNR, some of the normal data distributed around zero may be wrongly labeled as missing data. The training process requires a large amount of historical data, which is difficult for some problems. Also, the number of hyperparameters is too many to be optimized and demands for the expert experience.

In future work, further research effort is required to further improving the proposed algorithmic solution from two aspects. Firstly, the models can be further enhanced through the adoption of more sophisticated deep learning models. Also, the solution can be incorporated with a hybrid model that consists of multiple different machine learning algorithms. In addition, the proposed solution can be applied and validated for different time-series data in other application domains.

Data Availability

The Belgium load data used to support the findings of this study are available at <http://www.elia.be>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Science and Technology Project of State Grid Zhejiang Electric Power Co., Ltd. and Fundamental Research Funds for the Central Universities (Zhejiang University NGICS Platform).

References

- [1] S. Chattopadhyay, M. Mitra, and S. Sengupta, *Electric Power Quality*, Springer, Dordrecht, The Netherlands, 2011.
- [2] P. Gao, M. Wang, S. G. Ghiocel, J. H. Chow, B. Fardanesh, and G. Stefopoulos, "Missing data recovery by exploiting low-dimensionality in power system synchrophasor measurements," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1006–1013, 2016.
- [3] J. Sun, H. Liao, and B. R. Upadhyaya, "A robust functional-data-analysis method for data recovery in multichannel sensor systems," *IEEE Transactions on Cybernetics*, vol. 44, no. 8, pp. 1420–1431, 2014.
- [4] J. Stones and A. Collinson, "Power quality," *Power Engineering Journal*, vol. 15, no. 2, pp. 58–64, 2001.
- [5] L. Liu, D. Zhai, and X. Jiang, "Current situation and development of the methods on bad-data detection and identification of power system," *Power System Protection and Control*, vol. 38, no. 5, pp. 143–147, 2010.
- [6] S.-J. Huang and J.-M. Lin, "Enhancement of anomalous data mining in power system predicting-aided state estimation," *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 610–619, 2004.
- [7] R. Baldick, K. A. Clements, Z. Pinjo-Dzagal, and P. W. Davis, "Implementing nonquadratic objective functions for state estimation and bad data rejection," *IEEE Transactions on Power Systems*, vol. 12, no. 1, pp. 376–382, 1997.
- [8] M. Yang, L. Meng, D. Li et al., "Identification of abnormal data of photovoltaic power based on class 3σ ," *Renewable Energy Resources*, vol. 36, no. 10, pp. 1443–1448, 2018.
- [9] V. Aggarwal, V. Gupta, P. Singh, K. Sharma, and N. Sharma, "Detection of spatial outlier by using improved Z-score test," in *Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 788–790, Tirunelveli, India, 2019.
- [10] H. Yang, Y. Wang, S. Fotso Tagne, and Q. Huang, "Abnormal pre-detection algorithm based on time series," in *Proceedings of the 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, pp. 262–265, Newark, CA, USA, 2019.
- [11] Y. Huang, J. Xiao, Y. Li et al., "A new method to detect and identify bad data based on correlativity of measured data in power system," *Power System Technology*, vol. 30, no. 2, pp. 70–74, 2006.
- [12] M. Sohail Ibrahim, W. Dong, and Q. Yang, "Machine learning driven smart electric power systems: current trends and new perspectives," *Applied Energy*, vol. 272, 2020.
- [13] S. Wang, H. Chen, and Z. Pan, "A reconstruction method for missing data in power system measurement using an improved generative adversarial network," *Proceedings of the CSEE*, vol. 39, no. 1, pp. 56–64, 2019.
- [14] M. Yang, Y. Sun, G. Mu et al., "Data completing of missing wind power data based on adaptive neuro-fuzzy inference system," *Automation of Electric Power Systems*, vol. 38, no. 19, pp. 16–21, 2014.
- [15] S. Zhao and C. Wang, "Research and appreciation of the intelligent recovery of missing data in power system measurement," *Science and Technology Innovation Herald*, vol. 15, no. 18, pp. 96–98, 2018.
- [16] S. Quan and Y. Cao, "Interpretation method research application," *Science & Technology Information*, vol. 36, pp. 413–414, 2007.
- [17] P. Shi and L. Zhang, "A missing data complement method based on K-means clustering analysis," in *Proceedings of the IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pp. 1–5, Beijing, China, 2017.
- [18] V. Miranda, J. Krstulovic, H. Keko, C. Moreira, and J. Pereira, "Reconstructing missing data in state estimation with autoencoders," *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 604–611, 2012.
- [19] C. T. Concepción, S. L. Fernando, C. R. José et al., "A new missing data imputation algorithm applied to electrical data loggers," *Sensors*, vol. 15, no. 12, pp. 31069–31082, 2015.
- [20] C. K. Enders, *Applied Missing Data Analysis*, Guilford Press, New York, NY, USA, 2010.
- [21] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, Wiley, New York, NY, USA, 1958.
- [22] L. Li, Z. Wen, and Z. Wang, "Outlier detection and correction during the process of groundwater level monitoring base on pauta criterion with self-learning and smooth processing," in *Proceedings of the Asian Simulation Conference SCS Autumn Simulation Multi-Conference*, Berlin, Germany, 2016.
- [23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 1 (NIPS'12)*,

- pp. 1097–1105, Curran Associates Inc., Red Hook, NY, USA, 2012.
- [24] B. L. Yoon, “Artificial neural network technology,” *ACM SIGSMALL/PC Notes*, vol. 15, no. 3, pp. 3–16, 1989.
 - [25] M. Tropea and G. Fedele, “Classifiers comparison for convolutional neural networks (CNNs) in image classification,” in *Proceedings of the 23rd IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications (DS-RT ’19)*, pp. 310–313, IEEE Press, New York, NY, USA, 2012.
 - [26] K. Yanai, R. Tanno, and K. Okamoto, “Efficient mobile implementation of a CNN-based object recognition system,” in *Proceedings of the 24th ACM International Conference on Multimedia (MM ’16)*, pp. 362–366, Association for Computing Machinery, New York, NY, USA, 2016.
 - [27] M. Hu, H. Guo, and X. Ji, “Automatic driving of end-to-end convolutional neural network based on mobilenet-V2 migration learning,” in *Proceedings of the 12th International Symposium on Visual Information Communication and Interaction (VINCI’2019)*, pp. 1–4, Association for Computing Machinery, New York, NY, USA, 2019.
 - [28] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
 - [29] P. Vincent, H. Larochelle, Y. Bengio et al., “Extracting and composing robust features with denoising autoencoders,” in *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103, Helsinki, Finland, 2008.
 - [30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
 - [31] J. Masci, U. Meier, D. Ciresan et al., “Stacked convolutional auto-encoders for hierarchical feature extraction,” in *Proceedings of the 21st International Conference on Artificial Neural Networks*, pp. 52–59, Espoo, Finland, 2011.
 - [32] J. Sun, Y. Jin, and M. Dai, “Discussion on testing the mechanism of missing data,” *Mathematics in Practice and Theory*, vol. 43, no. 12, pp. 166–173, 2013.
 - [33] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, “An improved method to construct basic probability assignment based on the confusion matrix for classification problem,” *Information Sciences*, vol. 340–341, pp. 250–261, 2016.
 - [34] H. Huang, H. Xu, X. Wang, and W. Silamu, “Maximum F1-score discriminative training criterion for automatic mispronunciation detection,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, pp. 787–797, 2015.
 - [35] T. Yang, Z. He, D. Zhao et al., “FSOM neural network based on the network harmonic measurement missing data repair algorithm,” *Power System Technology*, vol. 44, no. 5, pp. 1941–1949, 2020.
 - [36] D. Pathak, “Context encoders: feature learning by inpainting,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, Las Vegas, NV, USA, 2016.

Research Article

The Abnormal Detection for Network Traffic of Power IoT Based on Device Portrait

Jiaxuan Fei ^{1,2}, Qigui Yao ^{1,2}, Mingliang Chen ³, Xiangqun Wang ^{1,2} and Jie Fan ^{1,2}

¹Global Energy Interconnection Research Institute Co., Ltd., Nanjing, China

²State Grid Key Laboratory of Information & Network Security, Nanjing, China

³State Grid Jiangxi Electric Power Co., Ltd., Ganzhou, JiangXi, China

Correspondence should be addressed to Jiaxuan Fei; 444965979@qq.com

Received 2 September 2020; Revised 8 October 2020; Accepted 10 November 2020; Published 24 November 2020

Academic Editor: Ting Yang

Copyright © 2020 Jiaxuan Fei et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The construction of power Internet of things is an important development direction for power grid enterprises. Although power Internet of things is a kind of network, it is denser than the ordinary Internet of things points and more complex equipment types, so it has higher requirements for network security protection. At the same time, due to the special information perception and transmission mode in the Internet of things, the information transmitted in the network is easy to be stolen and resold, and traditional security measures can no longer meet the security protection requirements of the new Internet of things devices. To solve the privacy leakage and security attack caused by the illegal intrusion in the network, this paper proposes to construct a device portrait for terminal devices in the power Internet of things and detect abnormal traffic in the network based on device portrait. By collecting traffic data in the network environment, various network traffic characteristics are extracted, and abnormal traffic is analyzed and identified by the machine learning algorithm. By collecting the traffic data in the network environment, the features are extracted from the physical layer, network layer, and application layer of the message, and the device portrait is generated by a machine learning algorithm. According to the established attack mode, the corresponding traffic characteristics are analyzed, and the detection of abnormal traffic is achieved by comparing the attack traffic characteristics with the device portrait. The experimental results show that the accuracy of this method is more than 90%.

1. Introduction

Power IoT (Internet of things) is to apply the Internet of things technology in the smart grid business, in power generation, transmission, substation, power distribution, and utilization, and so, on each link, the comprehensive deployment has edge information awareness, calculation ability, and management ability to execute the terminal device, effectively integrate power system infrastructure and communications infrastructure resources, and promote the operation of the enterprise operating the whole process of the whole scene perception, information fusion, and intelligent decision support, to raise the efficiency of utilization of electric power system existing infrastructure for the grid all the chain management to provide important technical support [1, 2]. With the wide application of Internet of things technology in the SGC (State Grid Co., Ltd.) to adapt

to the traditional industry and the trend of the Internet to accelerate convergence, the content associated terminal will increase by geometric series, through the sensor technology, communication technology, and computer technology to terminal access networks; it puts forward higher requirements on network security protection. At the same time, due to the particularity of information perception and transmission mode in the Internet of things, the transmission information of the Internet of things is easy to be stolen and replay. Traditional security measures can no longer meet the security protection requirements of the new Internet of things devices. Therefore, how to realize the security of the ubiquitous power Internet of things and build a full-scene security protection system that adapts to the ubiquitous power Internet of things has become an urgent problem to be solved by The State Grid Company.

The power Internet of things architecture consists of four logical levels from bottom to top: sensing, transport, platform, and application [3]. The sensor layer collects raw data from smart meters, sensors, handheld terminals, cameras, PCS, and other smart devices, which are the source of all data of power grid companies. The network layer transmits the data collected by the perception layer to the platform layer securely and reliably through the network communication technologies such as power communication network and wireless private network. The platform layer stores, clusters, and analyzes data to provide data support for the application layer. The application layer provides data to users for service [4–7]. In the perception layer, the power system's devices owned limited computing capacity and less storage. So, the traditional authentication method is not suitable, which will bring serious security problems. Communication methods and network protocols are complex, which makes network security protection more difficult. In the application layer, the interface of security responsibility is not clear, and there is a management gap. Power Internet of things lacks complete network security protection standards for power utilities. However, with the development of new services and the change of security situation, the extensive access of a large number of terminal equipment and users in the ubiquitous power Internet of things environment increases the network exposure surface, which brings severe challenges to the protection system characterized by boundary isolation. In addition, the ubiquitous power Internet of things gives birth to a large number of new business models, business interaction is more complex, and more flexible and accurate security policies and protection measures are urgently required.

Therefore, this paper takes terminal devices in the power Internet of things as the object, extracts features based on the traffic generated by their information exchange and constructs device portraits. Device portrait is the tagging of information, which presents the overall state of the device by describing the characteristics of a series of individuals. Through the device portrait, we can clearly and intuitively see the various feature dimensions of all the devices. The network anomaly detection technology can identify the attack behavior of illegal devices according to the device portrait and inform the system to intercept and deal with it in time. The information interaction between networks is carried by network traffic, and the behavioral characteristics of network attacks will naturally be reflected in the network traffic generated. Therefore, device portrait constructed according to device traffic is an efficient and real-time anomaly detection technology.

The simulation results show that the average accuracy of abnormal behavior detection can reach more than 90%. The rest of the article is structured as follows. We begin with an overview of the work related to flow anomaly detection in Section 2. In Section 3, we will introduce the main work of the scheme. In addition, in section 4, we introduce the simulation results and finally summarize them in Section 5.

2. Related Work

The premise of network anomaly detection technology is to understand the abnormal behavior of the network attack. Abnormal activities can be divided into three types: point exception, context exception, and collective exception. According to the attacker's objectives and activities, the attack is classified as the following four types: DoS (denial of service) attack, probing attacks, the user to the root (U2R), and remote to the user (R2U). In network anomaly detection technology, the representation of abnormal behavior is usually divided into two kinds: fraction and binary tag. According to [8], the current network anomaly detection technologies can be divided into four categories: based on classification, typical classification technologies include support vector machine, Bayesian network, and neural network. Based on statistical theory, chi-square test statistics are taken as the standard. Typical statistical theory-based methods include hybrid model, signal processing technology, and principal component analysis.

Since Denning first put forward the network intrusion detection model in 1980s, scholars at different times have put forward many network abnormal behavior detection methods with their own advantages by using the latest technology. At present, abnormal behavior detection methods mainly include four kinds: the most basic method based on statistical analysis, the method based on feature rules based on comparison and matching of data features and feature base, the method based on data mining with automatic analysis capability, and the most common method based on machine learning. Because machine learning algorithms can detect unknown patterns effectively, network anomaly detection based on machine learning has become the focus of research in recent years, for example, network abnormal behavior detection based on clustering, naive Bayes, or decision tree.

There are many algorithms in machine learning, and they have their own advantages and disadvantages. Therefore, scholars further study and innovate on the basis of previous studies, fuse multiple algorithms or improve them to design a new model, effectively make use of their respective advantages and avoid disadvantages to improve the effect of network anomaly detection. B. Senthilnayaki et al. [9] combined a genetic algorithm and support vector machine algorithm to propose a network anomaly analysis method, and the results showed that the detection accuracy of partial features extracted with genetic algorithm was higher than that of the support vector machine model trained with all features. Chang et al. [10] proposed a network anomaly detection method based on RF and SVM to address the low detection rate of network anomaly detection and found that the combination of feature extraction and machine learning could improve the detection rate. Gao et al. [11] proposed an adaptive integration model. By adjusting the proportion of training data, setting up multiple decision trees, and constructing MultiTree algorithm, the comparison test proved that the detection accuracy was improved, and it was found that the quality of data features was an important factor determining the detection effect.

Naseer et al. [12] studied the applicability of deep learning in network anomaly detection and implemented anomaly detection models based on different depth neural network structures, including convolutional neural network, autoencoder, and regression neural network. Tavoli [13] proposed a new intrusion detection method based on MLP neural network in view of the shortcoming that traditional anomaly detection methods cannot detect unknown anomalies in the network with high speed and complexity. Experimental results show that this method is superior to other methods in reducing false positives. Yong [14] proposed an intrusion detection algorithm based on the convolutional neural network. This network model has higher accuracy and detection rate than classical BP neural network, SVM algorithm, and deep learning algorithm DBN, which improves the classification accuracy of intrusion detection recognition. Zhang et al. [15] proposed an intrusion detection method based on deep learning, which uses a deep automatic encoder to compress unimportant features, extract key features, and build a model, and it uses the NSL-KDD data set to conduct tests to quickly and accurately identify attacks.

3. The Design of Scheme Architecture

The overall framework of this paper is mainly divided into two parts, as shown in Figure 1: the construction of interrupt device portrait and the detection of abnormal network access behavior of devices under specific attack scenarios.

Taking all the traffic generated by the terminal equipment as the object of study and comprehensively considering the physical layer, network traffic, and protocol behavior characteristics of the terminal equipment, the portrait of the terminal equipment was established. Based on the established device portrait and combined with the specific attack scenario, analyze and detect whether the network access behavior of the terminal is abnormal. The specific implementation route is shown in Figure 1.

3.1. Device Portrait. Based on the device data, utilizing tagging and taking the device as a unit, the data model is established to analyze the device data and extract the labels of each dimension of the device. The label set of the device is the device portrait constructed. Equipment data interpretation can be from two aspects: from the perspective of physical view, different devices have different electronic components, in which the emission of the electromagnetic wave from different devices also are different. These characteristics include carrier frequency offset of the baseband's steady-state responses, synchronization signal correlation value, baseband I/O of offset, signal demodulation signal amplitude and phase error, etc. The value of these characteristics is unique to the different devices to be treated as the fingerprints of the devices. On the other hand, it starts from the network layer, which is mainly reflected in the network traffic generated by devices during the running time. Due to different functions and applications, the performance on the TCP/IP protocol stack is also different. According to these

two types of characteristics, useful data can be extracted from them to form a tag set using tagging. Then, based on the machine learning algorithm, exclusive portraits can be constructed for different devices to identify legitimate and illegal users.

Portrait refers to the digitization and labeling of information. A series of features that can represent the device are logically combined in a certain way to form a proprietary portrait of the device. The core work of the portrait is to find as many features that can represent the device as possible and as much as possible, to fuse multidimensional features, and to generate a device portrait by establishing a mathematical model to analyze device data.

3.2. System Model. Device portrait is a way to present the multidimensional description of device features. In this paper, the portrait of the terminal device contains two core contents, such as basic attributes and access behavior attributes, as shown in Figure 2. Basic properties include the terminal's IP, MAC address, and machine name. Access behavior attribute is composed of physical characteristics of a terminal device, network traffic characteristics, and protocol behavior characteristics.

Basic properties can be obtained from network traffic packets. The physical layer characteristics of access behavior attributes include transient signal fingerprint characteristic vector of individual characteristics of communication devices.

The characteristics of the physical layer include the constellation locus of BPSK, QPSK, OQPSK, and MSK modulation signals in the receiving end baseband and the time domain and frequency domain characteristics of the time domain waveform. The constellation trajectory chart is a means to measure the emitted signal itself and its change law. The nonlinear response of the transmitter amplifier, the response of the filter, and other linear and nonlinear interference factors will be reflected in the changing trajectory of the constellation trajectory map. A constellation chart provides a more comprehensive measure of the characteristics of the received signal. After receiving the oversampled baseband signal, the receiver can preprocess the signal simply. Preprocessing is mainly to normalize the energy of the signal. After preprocessing the received signal, the signal is sent to the I/Q two-channel delay device. The delay can choose the same and different delays for both I/Q channels. The choice of I/Q two-channel signal delay is mainly determined by the judgment of signal modulation. After that, the system performs differential processing on the signal, and the stable and clear constellation trajectory diagram can be drawn on the complex plane.

Network traffic features include target IP address distribution (number), magnitude distribution (mean and variance) of upstream and downstream traffic, duration distribution (mean and variance) of upstream and downstream traffic, and network flow order. The distribution of the target IP address, the size distribution of upstream and downstream traffic, and the duration distribution of upstream and downstream traffic can be extracted from the

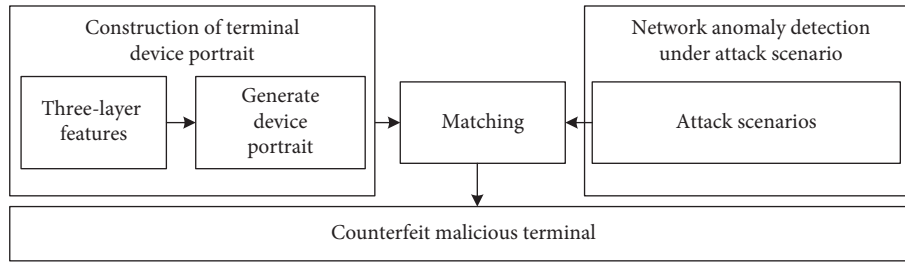


FIGURE 1: Implementation roadmap of abnormal network access behavior detection based on device portrait.

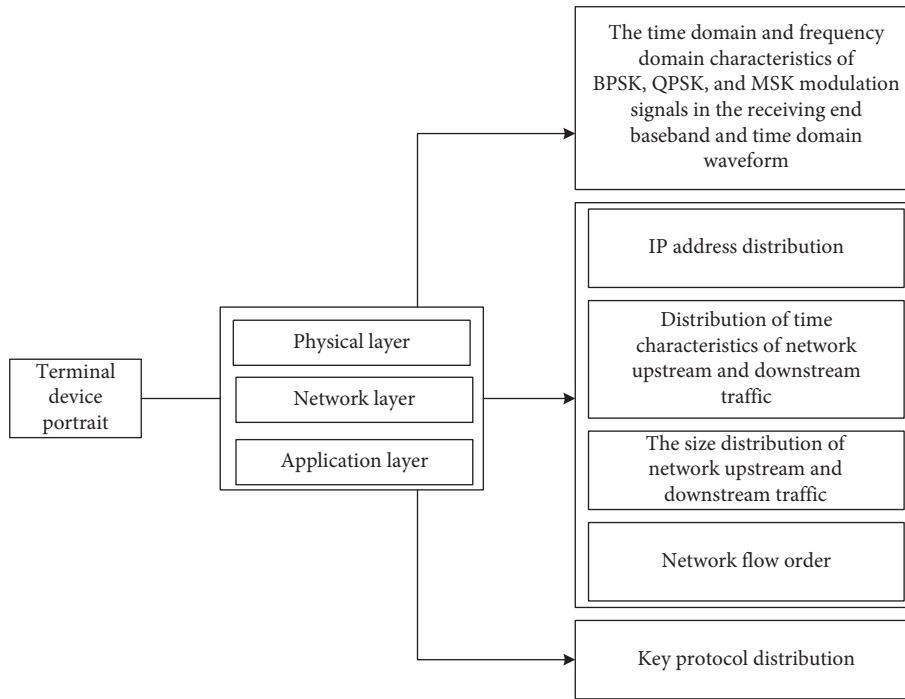


FIGURE 2: Portrait content of terminal equipment.

network flow order and calculated statistically. Statistics of such characteristics is convenient for rapid detection. There are also Timestamp field, MSS, WScale, SYN, FIN, ACK, PSH, URG, and RST in the TCP header. Version number, IHL, TTL, DF, TOS, Protocol field value, source port number (Sport), destination port number (Dport), and so forth are shown in Tables 1 and 2.

The characteristics of the application layer include protocol key fields, writing all the protocol keywords that appear into the document, and reflecting the protocol characteristics in the form of files, without specific analysis of the business type. The characteristics of the service layer can be determined according to the parallel parsing results of the protocol message.

3.3. *Abnormal Network Access Behavior Detection of Devices under Specific Attack Scenarios.* Time range T can be set for the construction of the device portrait, and the eigenvalue in time T is the current portrait of the device. Based on device portrait, abnormal network access behavior of terminal devices of power Internet of things can be further detected.

TABLE 1: TCP and IP header field value.

| | |
|-----|--|
| TCP | Timestamp, MSS, WScale, SYN, FIN, ACK, PSH, URG, RST |
| IP | Version, TOS, IHL, TTL, DF, Protocol, Dport, Sport |

TABLE 2: Application layer protocol.

| | |
|-----------|---|
| TCP-based | http, https, smtp, ssh, ftp, lpd rtsp, telnet, raw |
| UDP-based | snmp, onvif, dns, ntp, mdns, ssdp, icmp, igmpv3, nfs, dhcp, tftp, pop |

Specifically, this project intends to analyze the portrait of terminal devices under specific attack scenarios and determine abnormal network access behaviors, to realize the fast and accurate detection of forgery and malicious terminals. The specific process is as follows:

Step 1. Establish attack modes. Referring to the network attack chain model, the process of the attack is summarized and the attack mode is established. Because the network attack chain model is mainly aimed at Internet attacks and the whole service is ubiquitous in the power Internet of things, there are special attacks, such as fake terminals and then access to background applications. Therefore, this project considers specific attack scenarios and plans to establish a common attack mode for the power Internet of things. Specifically, in addition to the normal stages of investigation and weaponization included in the network attack chain model, special stages such as terminal forgery and abnormal execution of normal instructions will be added, to model such special attacks as using forgery terminals to access specific services and then destroying business systems through normal instructions.

There are various types of network attacks, and there are corresponding attacks for each level. It is not practical to use a network anomaly detection model for all attacks. TCP-IP architecture is the infrastructure of today's Internet; many network attacks are aimed at TCP, IP layer, affecting the normal process of the target host. In this paper, SYN denial of service attack, TCP port scanning, and IP attack are selected to verify the detection performance of the system, among them the SYN denial of service attack. In the three-time handshake process based on TCP connection, the attacker sends SYN request message to the target, and the target host will assign resources after receiving the message, send the response message, and wait for the attacker to reply. The attacker does not respond to the target, making the target in a waiting state. Through a large number of SYN request packets, the resource of the target host can be exhausted. TCP port scanning: when the corresponding port of the target host receives the TCP connection request, if the port is open, the TCP ACK message is sent back to establish the connection, and if the port is not open, the TCP RST message is sent to inform the sender that the port is not open. By sending requests to all ports of the target in turn, detecting the received response can tell which ports of the target are open for the next attack. The IP address of the target host is "192.168.2.102". The target port is traversed and selected in (1,65535) to parse the received message. If the TCP layer flags = 18, the port is open; otherwise, it is closed. Sharding IP packet attack: when an IP packet is too large, it will be shared by the IP layer. The flag MF of the sharded packet tells the receiver that the packet has been sharded. The target host will receive these packets into the cache waiting for the arrival of subsequent packets and merge them. Sending a large number of sharding messages to the target artificially will deplete the cache resources of the target host. The IP address of the target host is "192.168.2.102", the target port is 80, the source address is any available IP

in the network segment, and the source port is any port in the interval. Each IP address will be traversed to send five different ids, and each ID will send five IP messages with different slice offsets.

Step 2. Analyze the traffic characteristics of the established attack mode. For each attack step in the established attack mode, the corresponding traffic characteristics are analyzed. The flow characteristics analyzed include physical layer characteristics, such as time domain and frequency characteristics of the communication channel. Network traffic characteristics and network flow order of attack traffic are constructed. Protocol layer characteristics build the attack traffic behavior model diagram.

Step 3. Compare and distinguish attack traffic characteristics and device portraits based on the cluster analysis algorithm and the nearest neighbor set selection. If similar traffic characteristics are found, the terminal is judged to be a counterfeit or malicious terminal.

Clustering technology is a widely applied unsupervised machine learning algorithm, which combines the k-means algorithm and improved collaborative filtering algorithm to realize network anomaly detection technology based on device image. In the process of dividing a set of objects into different classes, the same class of data should have similar characteristics, and the data characteristics in different classes should be highly different. Data sets containing normal and abnormal data form clusters of various sizes, and the smaller and sparse ones are generally considered as abnormal. After the completion of clustering, the similarity between the observed data and each cluster center is calculated, and the cluster with a high similarity is selected as the nearest neighbor set. If the similarity between the abnormal cluster and the normal cluster is higher than that between the abnormal cluster and the normal cluster, or the similarity between the normal cluster and the abnormal cluster is less than the established threshold, it is considered as an anomaly. The detailed process of clustering is as follows:

- (1) Randomly select K objects from n objects as the initial clustering center
- (2) Calculate the difference between all objects and the central object according to the mean value of each cluster object, and divide the elements into the cluster with the lowest difference
- (3) The mean value of each cluster with changes was calculated again
- (4) Repeat steps b and c until no change occurs in each cluster
- (5) Output results

In the process of selecting the nearest neighbor, the similarity is calculated through the weighted idea of various similarity balance factors [12], and the selected similarity measurement factors mainly include feature matrix

similarity and feature similarity of device portrait; then, the similarity calculation formula in this section is

$$\sin(u, v) = \alpha \sin_r(u, v) + \beta \sin_e(u, v), \quad (1)$$

where $\sin_r(u, v)$ is used to describe the similarity of different features between the device u to be observed and the existing device; $\sin_e(u, v)$ is used to describe the similarity of the image of the device to be observed and the existing device; and α, β in turn are used to describe the corresponding weights. After the similarity is obtained, the results with the largest similarity are taken as the nearest neighbor. The following is the calculation of equipment feature similarity. Assuming there are m features, the m features of the observation equipment u and the existing equipment v are used and described by Q_u and Q_m in turn. The characteristic similarity of equipment v and u is the similarity between Q_u and Q_m . The calculation formula is as follows:

$$\sin_r(u, v) = \frac{|Q_u \cap Q_m|}{|Q_u \cup Q_m|}, \quad (2)$$

where $Q_u \cap Q_m$ is used to describe the characteristic quantities of the same value shared by devices u and v . $Q_u \cup Q_m$ is used to describe the number of uncharacteristic rights in common. The device portrait similarity is described by the Pearson correlation coefficient. The Pearson correlation coefficient is valued in the range of $[-1, 1]$. The higher the absolute value of the Pearson correlation coefficient, the stronger the correlation. When the Pearson correlation coefficient is 1, the similarity is the highest and positively correlated. When the Pearson coefficient is -1 , the correlation is considered to be completely negative. When the Pearson correlation coefficient is 0, no relationship is considered. The similarity between u and v is measured by Pearson's correlation coefficient as follows:

$$\sin_e(u, v) = \frac{\sum(r_u - \bar{r}_u)(r_v - \bar{r}_v)}{\sqrt{(r_u - \bar{r}_u)^2} \sqrt{(r_v - \bar{r}_v)^2}}, \quad (3)$$

where r_u is used to describe the characteristic value of equipment u and \bar{r}_u is the average value of the characteristics of multiple traffic flows of the equipment. r_v is the existing cluster, namely, the single eigenvalue of the device portrait of a certain device, and \bar{r}_v is the average characteristic value in the device portrait.

According to the characteristics of the collaborative filtering algorithm, the device portrait similarity is optimized to prevent the nearest neighbor from being found in the whole object space. Since $r_u - \bar{r}_i$ and $r_v - \bar{r}_j$ are not defined as nonnegative, there are

$$(r_u - \bar{r}_u)(r_v - \bar{r}_v) \leq 4[\max(\bar{r}_u, R - \bar{r}_u, \bar{r}_v, R - \bar{r}_v)], \quad (4)$$

where R is used to describe the value of the most similar feature.

In particular, specific attack stages can be detected according to different matching results. If the physical layer fingerprint features are abnormal and the network and

business contents are normal, it indicates that the attacker may change the core equipment into his equipment to prepare for the subsequent attack. If the physical layer fingerprint characteristics are normal, but the network traffic characteristics are abnormal, then the business content is normal, indicating that the attacker is contacting the background server, receiving instructions or updating the attack code, and so forth. To prepare for the attack. If the physical layer and network traffic characteristics are normal and the business layer characteristics are abnormal, it indicates that the attacker is carrying out a business attack to complete the attack preparation.

4. Experimental Analysis

4.1. Simulation Experiment Environment. The experiment was conducted on the Pycharm open-source platform, which can build various machine learning algorithms, including the application of the clustering algorithm and the collaborative filtering algorithm. Build a recommendation engine with tools provided by the platform.

In this paper, the bypass monitoring traffic method is adopted to implement network data traffic through bypass monitoring which includes bypass detection, data acquisition, network analysis, and information extraction. Specific scheme of data acquisition based on bypass detection: set up bypass monitoring on the data switch, regularly collect data traffic packets of devices accessing network services and internal resources, and extract network protocol and business information related to terminal devices. Its architecture block diagram is shown in Figure 3.

By mirroring on the network switch port, bypass monitoring is set to filter the detected network packets according to the corresponding MAC address, to obtain the packets related to the device terminal. In data communication, the data packets generated by each network activity of the terminal equipment can be regarded as the process of data interaction between two nodes. This passive device traffic acquisition method is less aggressive and invasive, has no restrictions on the type of device, and is easy to operate in practice, which is also the reason why passive device traffic acquisition is selected in this paper.

There is another problem that needs to be considered. The terminal equipment will show different network behaviors with the change of the execution task, so the network traffic generated by the equipment is of great uncertainty, which will affect the accuracy of our portrait generation. Faced with this problem, we divide the entire running process of the terminal device into two parts: the startup phase and the service phase. The process from power-on to complete the hardware and software configuration in the startup stage: the service stage is the stage in which the device performs functional tasks after completing various configurations. In equipment beginning to perform a task, its network behavior is highly susceptible to the influence of network environment, such as the network administrator to configure the network environment changes and the communication terminal entity flow changes, and this kind of change in a certain period of time is needed to show some

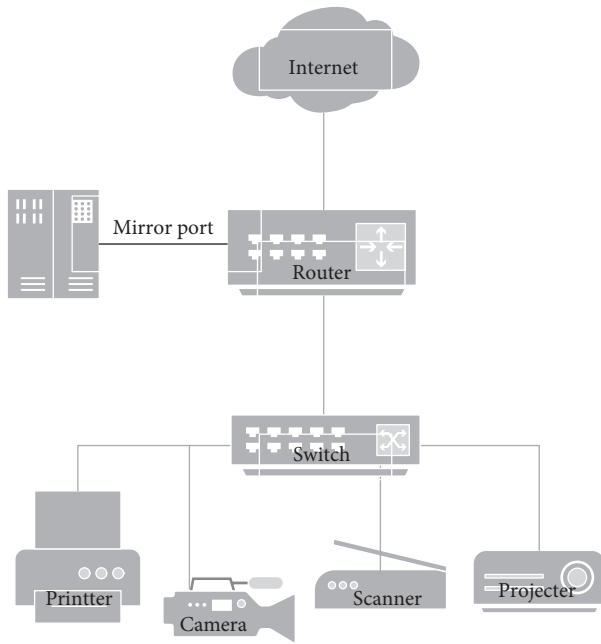


FIGURE 3: Data acquisition architecture diagram.

kind of rule and has the characteristics of uniqueness, but this time cycle length is not fixed. For example, the periodic behavior model of access point devices can be analyzed by analyzing individual data packets, which is not feasible in practical operation. We can use the characteristics of the startup phase to solve this problem. Due to its single function and limited resources, many configurations of intelligent devices in the Internet of things will be fixed in the hardware or software settings, that is, the configuration in the startup stage. The transition from power failure to the normal state of the device has a strict loader, which provides us with a stable flow window starting time. Launch complex Linux operating system. For example, firstly, the basic input-output system (BIOS) is checked to confirm the hardware status is normal; secondly, load the disk, and then load the master boot record (MBR) of file system's; Thirdly, read the content of the code from the MBRboot, which is the bootstrap program. The bootstrap program contains the code to load the system's kernel, start the initialization process, perform the script from different levels, etc.; Finally, the whole booting procedure is completed by the bootstrap program of the operating system loads. The startup process takes a relatively fixed time, so the time window experienced in the startup stage is relatively stable. In addition, most of the installation systems of smart devices in the Internet of things are not heavyweight systems, so the startup time is about 3 minutes or less. Therefore, we choose to obtain the traffic packets during the startup period of the device to construct the device portrait.

In the attack module, this paper constructs a system consisting of two hosts, making one host the attacker and the other the target host to receive the attack. The data acquisition module collects attack data and traffic generated by normal communication, collects and saves network attack data and normal data together, and analyzes packets. Feature

extraction module is responsible for feature extraction and processing, changing network data from unordered to TCP connection, and writing code to calculate the feature value of each connection from these fields. After feature extraction, feature processing should be carried out to make features meet the requirements of the machine learning algorithm. The three attacks generated in this article are all based on the Scapy library. Scapy is used to set the field value of the network protocol, which can generate the required packets and form the network attack.

4.2. Determine the Weight Value of Similarity. The weights α and β in the similarity calculation of attack flow and device portrait are in line with $\alpha + \beta = 1$. According to the actual data and the least square fitting method in linear programming, α and β are adjusted to obtain the optimal result. The following are three different cases of $\alpha = 0.65, \beta = 0.35$, $\alpha = 0.55, \beta = 0.45$, and $\alpha = 0.35, \beta = 0.65$ to compare the accuracy rate and recall rate. The success rate is the proportion of the total weight of the successful verification match of the attack device. The recall rate is the proportion of the number of successful matches in the candidate set to the actual number of received matches in the experiment.

It can be seen from Table 3 that, in the case of increase, the accuracy rate gradually increases and is the highest in the case of 0.65. By synthesizing all the elements, the similarity analysis in cluster analysis takes time.

4.3. Comparison of Different Algorithms. After determining the parameters of cluster analysis, the device portrait is constructed by comparing different machine learning algorithms. In the experiment, the selected machine learning algorithms include logistic regression, decision tree, and random forest. The algorithm which is most consistent with device portrait and anomaly detection is selected. The results are as in Table 4.

As can be seen from Table 4, the cluster analysis algorithm has the highest accuracy of 91.2% in TCP port scanning, while the random forest algorithm has the lowest accuracy of 88.7%. In IP sharding attack, the accuracy rate reaches 90.9% in the clustering algorithm and 80.3% in the decision tree. In the SYN denial-of-service attack, the logistic regression algorithm had 90.5% accuracy, and the second highest was cluster analysis, with a difference of only 0.2%.

According to the results in Table 4, among the above four algorithms, the device portrait constructed by the clustering analysis algorithm has a high accuracy rate for the identification of device anomalies in the network, meeting the requirements of the system.

4.4. The Influence of Features on Anomaly Detection. The device portraits constructed by different features are also different. To observe the influence of device portraits on anomaly detection, different features are deleted, respectively, to observe the difference of experimental results.

Figures 4–6, correspond to the influence of device portrait on detection results in three attack modes: TCP Port

TABLE 3: Weight analysis results.

| Weighted value | Recall (%) | Precision (%) |
|-------------------------------|------------|---------------|
| $\alpha = 0.65, \beta = 0.35$ | 86.2 | 91.2 |
| $\alpha = 0.55, \beta = 0.45$ | 89.3 | 75.9 |
| $\alpha = 0.35, \beta = 0.65$ | 61.2 | 53.1 |

TABLE 4: Comparison of the accuracy of different attack types and different algorithms.

| Algorithm | TCP port scan (%) | SYN denial (%) | IP sharding attack (%) |
|---------------------|-------------------|----------------|------------------------|
| Cluster analysis | 91.2 | 90.3 | 90.9 |
| Logistic regression | 89.5 | 90.5 | 90.5 |
| Decision tree | 89.1 | 88.3 | 80.3 |
| Random forest | 88.7 | 78.5 | 85.8 |

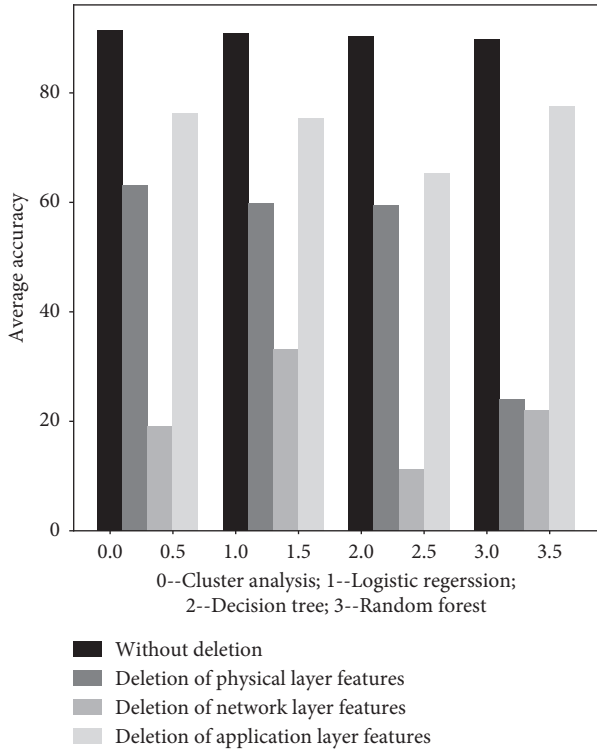


FIGURE 4: Influence of device portrait on port scanning.

Scan, SYN denial Attack, and IP sharding attack. The vertical axis represents accuracy, and the horizontal axis represents four algorithms from left to right: clustering, logistic regression, decision tree, and random forest. The first cylinder represents the result before deleting the feature, the second represents the accuracy of deleting the physical layer feature, the third represents only deleting the network layer feature, and the fourth represents the result of deleting the application layer feature.

As can be seen from Figures 4–6, after the deletion of network layer features, the system performance degrades significantly, followed by the physical layer and finally the application layer. It can be seen from the above that the

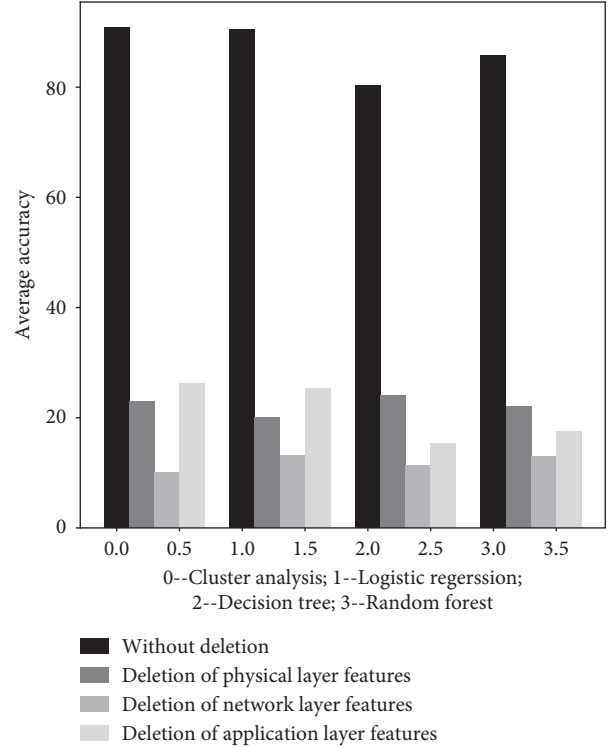


FIGURE 5: Influence of device portrait on SYN denial attack.

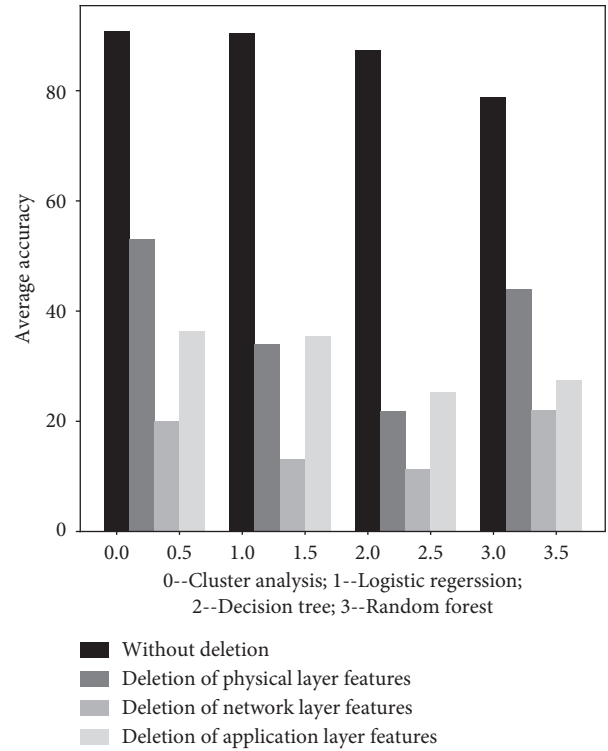


FIGURE 6: Influence of device portrait on IP sharding attack.

detection of abnormal network traffic by the machine learning algorithms is largely dependent on the perfection of eigenvalue selection. If some key features are removed, the

performance of the model will be significantly reduced. For the SYN denial attack, the performance of the four algorithms declines after removing several features, and the characteristics have the most obvious impact on the SYN denial attack. After the TCP port scanning attack is removed, clustering analysis and logistic regression can maintain high detection performance, while other algorithms decline. After the SYN denial of service attack was removed, the four algorithms all showed a certain degree of decline, and the random forest and cluster analysis performed slightly better. In the future upgrade, to enable the system to make more accurate judgment of network attacks, feature extraction needs to be increased and improved.

Device portrait is a description of a series of features of the device. The accuracy of device portrait to detect abnormal network traffic largely depends on the integrity of the portrait, that is, the integrity and accuracy of selected feature values. If some key features are removed, the monitoring performance of the system will be greatly reduced.

5. Conclusion

In this paper, we propose to construct device portraits for terminal devices in the power Internet of things and detect abnormal traffic in the network according to device portraits, to protect the security of the Internet of things to a certain extent. The experimental results show that the accuracy of this method is more than 90%. Due to resource constraints, we collected limited terminal equipment and traffic data and were unable to conduct large-scale testing. In the following work, we will collect more brushes and constantly improve the construction of the device portrait.

Data Availability

The experimental data source of this paper and the actual production and operation data of State Grid Corporation of China are only available on the company's internal network.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Science and Technology Project of State Grid Corporation of China (Grant no. 5700-201958466A-0-0-00): "End-to-End Security Threat Analysis and Accurate Protection of Ubiquitous Power Internet of Things."

References

- [1] C. Tian, *Application of Homomorphic Encryption in Block Chain Data Security of the Internet of Things Network Security Technology and Application*, vol. 3, pp. 34–36, 2018.
- [2] J. Liang, D. E. N. G. Yurong, L. Guo et al., "Research and application of remote monitoring for power transmission and transformation facilities based on satellite Internet of things," *Electric Power Construction*, vol. 34, no. 9, pp. 6–9, 2013.
- [3] L. Ling, L. Shancang, and Z. Shanshan, "QoS-aware scheduling of services-oriented internet of things," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1497–1505, 2014.
- [4] W. Zou, J. Chen, X. Weng et al., "The security analysis and countermeasure of power Internet of things," *Electric Power Information and Communication Technology*, vol. 12, no. 8, pp. 121–125, 2014.
- [5] J. Lü, W. Luan, R. Liu et al., "Architecture of distribution Internet of things based on widespread sensing & software defined technology," *Power System Technology*, vol. 42, no. 10, pp. 3108–3115, 2018.
- [6] C. Wu, *Security Basis of Internet of Things*, pp. 55–56, Science Press, Beijing, China, 2013.
- [7] R. Roman, J. Y. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed Internet of things," *Computer Networks*, vol. 57, no. 10, pp. 2266–2279, 2013.
- [8] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.
- [9] B. Senthilnayagi, K. Venkatalakshmi, and A. Kannan, "Intrusion detection using optimal genetic feature selection and SVM based classifier," in *Proceedings of the 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, pp. 1–4, IEEE, Chennai, India, March 2015.
- [10] Y. Chang, W. Li, and Z. Yang, "Network intrusion detection based on random forest and support vector machine," vol. 1, pp. 635–638, in *Proceedings of the IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC)*, vol. 1, pp. 635–638, Institute of Electrical and Electronics Engineers, Guangzhou, China, July 2017.
- [11] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *Institute of Electrical and Electronics Engineers*, vol. 7, pp. 82512–82521, 2019.
- [12] S. Naseer, Y. Saleem, S. Khalid et al., "Enhanced network anomaly detection based on deep neural networks," *Institute of Electrical and Electronics Engineers*, vol. 6, pp. 48231–48246, 2018.
- [13] R. Tavoli, "Providing a method to reduce the false alarm rate in network intrusion detection systems using the multilayer perceptron technique and backpropagation algorithm," in *Proceedings of the 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, IEEE, Tehran, Iran, March 2019.
- [14] L. Yong and Z. Bo, *An Intrusion Detection Model Based on Multi-Scale CNN[C]//2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference*, pp. 214–218, ITNEC, Chengdu, China, 2019.
- [15] C. Zhang, F. Ruan, L. Yin et al., "A deep learning approach for network intrusion detection based on NSL-KDD dataset," in *Proceedings of the IEEE 13th international conference on anti-counterfeiting, security, and identification (ASID)*, pp. 41–45, IEEE, Xiamen, China, July 2019.