IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 19, 2020, accepted July 2, 2020, date of publication July 9, 2020, date of current version July 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3008255

Wireless Sensor Network Fault Sensor Recognition Algorithm Based on MM* Diagnostic Model

WEIXIA GUI, QIAN LU, MEILI SU, AND FULAI PAN
School of Computer and Electronics Information, Guangxi University, Nanning 530004, China

Corresponding author: Weixia Gui (wxgui@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61862003 and Grant 61862004, and in part by the Ph.D. Scientific Research Foundation of Guangxi University under Grant XBZ200171.

ABSTRACT Wireless Sensor Network (WSN) as one of the representatives of the Internet of Things technology has also received much attention. To accurately diagnose fault sensor nodes, a fault diagnosis method based on fireworks algorithm optimization convolutional neural network algorithm is proposed. The weights and biases of the convolutional neural networks are optimized by using the self-regulating mechanism of global and local searching ability of fireworks algorithm. So the problem of convolution neural network in extreme judgment and limited convergence speed is solved, to effectively realize the fault diagnosis of the WSN. Simulation experiments show that this algorithm has higher fault diagnosis accuracy than other classic WSN fault diagnosis algorithms.

INDEX TERMS Convolution neural network, fault diagnosis, fireworks algorithm, MM* model, wireless sensor network.

I. INTRODUCTION

In the context of the development of big data and the Internet of Things, WSN has gradually become a research hotspot in various fields, especially in the fields of military warfare, environmental monitoring and forecasting, security monitoring, smart home, and health care. Sensor node failures are usually caused by the hardware failure of the sensor itself and the poor deployment environment. This can lead to frequent changes in the topology of the WSN, communication errors, and network separation. Therefore, timely diagnosis and removal of WSN faults, and improving the reliability and life cycle of WSN operation are the preconditions to ensure that WSN monitoring system can complete the specified tasks. In another words, WSN fault diagnosis is critical to maintaining the quality of the network.

Because of its great advantages in the application field, WSN has attracted wide attention from all walks of life, and various WSN fault diagnosis algorithms have been proposed one after another. The way of base station diagnosis is a common WSN diagnostic method. In 1967, Preparata, Metze and Chien proposed a system-level diagnostic model for interconnection–PMC model [1], in which the processing module tests other modules, using the results of the test to determine the state of the system. Subsequently, diagnostic models such as the BGM model [2], Chwa & Hakimi model [3], Malek model [4], MM model [5] and MM* model [6] were successively proposed. These diagnostic models are also suitable for WSN. Two sensor nodes that can communicate can be considered as two nodes with physical connections. De Paola [7] proposed an adaptive distributed Bayesian method for detecting outliers in data collected by WSN. Zhang and Yuan [8] proposed a new method for fault diagnosis of WSN clusters based on distributed energy-efficient clustering algorithm and neighbor collaboration strategy. The algorithm divides the entire network into multiple clusters, distributes fault management to each cluster area, and adopts an improved neighbor collaborative diagnosis strategy in the cluster. But the improvement effect of these two methods of diagnostic accuracy is not obvious. Based on the particle swarm optimization (PSO) classification method, Swain and Khilar [9] proposed a real-time soft fault diagnosis model for WSN. The proposed model is divided into three phases, such as initialization, fault

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Tucci.

identification, and fault classification phases, to diagnose composite faults in the sensor network. Subsequently, they proposed a heterogeneous fault diagnosis protocol for WSN [10]. However, the false alarm rate of these two methods is slightly higher when the node failure rate is higher than 50%. Gao *et al.* [11] proposed a distributed filtering scheme for dealing with soft fault detection of nonlinear stochastic systems with WSN. Cheng [12] proposed a sensor network fault detection mechanism based on support vector regression and neighborhood coordination. The fault prediction model was established by using a support vector regression algorithm. Then, the node state is determined by testing each other between reliable neighbor nodes. Although these methods improve the diagnostic accuracy to some extent, they increase the communication between sensor nodes. He *et al.* [13] proposed a new fault diagnosis method based on belief rules for WSN, which can detect and correct sensor node faults in time and improve the accuracy of sensor data fusion. But that requires a lot of historical data. Mengying *et al.* [14] proposed a wireless sensor fault diagnosis algorithm based on neighbor node and neighbor node data for the problem of low detection accuracy when the fault node rate is higher than 50% in WSN. In recent years, researches on fault diagnosis methods based on deep learning have become more and more popular, and many fault diagnosis methods based on neural network have emerged [15]–[17]. Their diagnostic accuracy and false alarm rate are much better than traditional methods, which makes fault diagnosis of WSN based on the neural network to be a future research trend.

Convolutional Neural Networks (CNN) [18] is a type of feed-forward neural network with convolutional computation and deep structure. It is one of the representative algorithms of deep learning. However, the traditional convolutional neural network learning method uses the steepest descent algorithm to learn, and the learning performance of the steepest descent algorithm is greatly affected by the initial weight and bias selection of the neural network. If the initial weight and bias selection are not good, the training process is easy to fall into local optimum. Considering that the fireworks algorithm has strong self-regulation mechanism of global search ability and local search ability [19], and fireworks algorithm can effectively solve the problem of system-level fault diagnosis [20], this paper introduces the fireworks algorithm into the convolutional neural network model and proposes a new WSN fault diagnosis method–FWA-CNNFD (Fireworks Algorithm-Convolutional Neural Networks Fault Detection). Firstly, according to the shortcomings of the traditional fireworks algorithm, the fireworks algorithm is improved appropriately. Then the initial weights and biases of the multi-group convolutional neural networks are generated. The combination of the weights and biases of each group is used as the initial population of the fireworks algorithm. Through explosion operator, mutation operator, displacement operation, mapping rule, and selection strategy, the optimal weight and bias are obtained. Finally, the optimal weight and bias are used as the initial weight and bias of the convolutional neural network to construct the wireless sensor fault diagnosis model. By combining the fireworks algorithm with the convolutional neural network, the wireless sensor fault diagnosis efficiency is greatly improved.

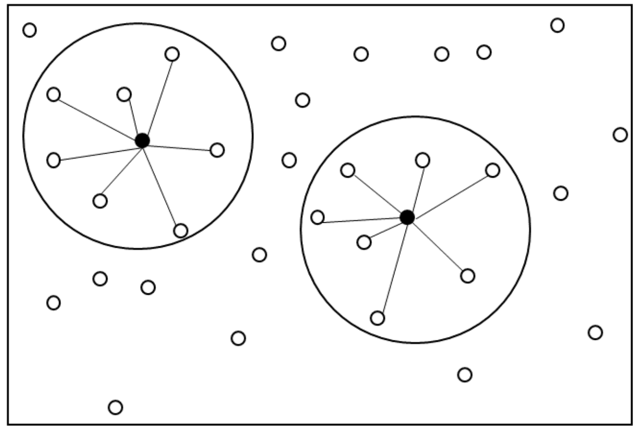


FIGURE 1. Typical WSN.

The rest of the paper is organized as follows. Preparation knowledge is discussed in Section II. In Section III, we improve the neural network based on the fireworks algorithm. The implementation and analysis of FWA-CNNFD algorithm are described in Section IV. Finally, the conclusion and the direction of future work are given in Section V.

II. PREPARATORY KNOWLEDGE

A. WSN FAULT DIAGNOSIS MODEL

Wireless sensors are generally deployed in environments with harsh environments and sparsely populated environments. From the beginning of the deployment, nodes are almost unmanaged, so WSN failures are more frequent and unpredictable than traditional wireless networks. If a component has hardware or software errors, it will lead to failure [21]. So, the sensor network is inevitable faulty, and what we have to do is to determine which nodes in the network have failed.

The WSN consists of several wireless sensor nodes in a self-organizing manner (Figure 1) [22]. The wireless sensor consists of four basic modules: sensor, CPU, wireless communication and power supply. Nodes can communicate within a certain communication range and can transmit and exchange information with each other. The sensor also has a simple calculation function. In Figure 1, the small circles represent sensor nodes, and the black nodes represent the nodes that initiated the communication request. The line segment between the two points indicates that the two sensor nodes can communicate, and the large circle represents the communication range of the black node.

With the development of WSN technology, various fault detection, and fault diagnosis models have been proposed. Malek comparative model was first proposed by Malek. Under this model, the diagnostic test is executed by the node that acts as the arbitrator. The arbitrator sends the same task and input to the two nodes that need to be tested, and then the arbitrator compares the operation results of the two nodes for the given task and input. This centralized troubleshooting approach has proven to be effective and accurate, but this approach is not suitable for large-scale networks. The main reason is that such a cost will be very high. The base station or center node collects node information in a centralized way, and the fault identification requires a high level of equipment, so this method does not apply to large-scale networks. Also, this method also causes the node near the base station or the central node to quickly consume energy, and the lifetime is reduced, thereby causing the network to be disconnected. Therefore, for the defects of the centralized model, Sengupta and Danbura proposed a distributed comparison model, the MM* model. In this model, the operation results of each node are compared. This operation is not performed by a single arbiter, but each node can compare the operation results of any two nodes adjacent to it. By collecting all the comparison results, we can judge the status of each node in the system. Nodes in the network can independently judge the state of the network, and the node makes decisions under the supervision of the base station. If the node can make more decisions, the less information that needs to be fed back to the base station and other centers for judgment, thus effectively reducing the extra energy overhead and extending the life of the network. And this diagnostic framework can make the diagnosis of large-scale dense networks easier. Therefore, the MM* model has been widely applied and studied.

Wireless sensor network faults can be divided into hard faults and soft faults. Hard fault means that the sensor node cannot collect data or cannot communicate with other nodes due to insufficient power or other reasons. Soft fault refers to the normal data collection and communication function of the sensor node, but there is a certain error between the collected data and the true value. Soft faults can be subdivided into the following four types: (1)Permanent faults. The data sensed by the node will not change with time, and the sensed data is a fixed value; (2)Temporary faults. The node perceives that the data is abnormal during a certain period of time, but perceives that the data is normal at other times; (3) Transient faults. At a certain moment, the node feels wrong, and at other times the sensed data is normal; (4) Offset faults. Perceptual data is larger or smaller than a fixed value at every moment than real data. This article assumes that the sensor node is a permanent soft fault.

B. MM* MODEL

Under the MM* model, an undirected graph $G(U, E)$ can be used to represent the test relationship between sensor nodes in a WSN. One vertex $u \in U$ in the graph represents one sensor node in the system, and the edge $(u_i, u_j) \in E$ represents the sensor node u_i and u_j can communicate with each other. Node w assigns the same task to its two adjacent nodes u, v and then feeds the test result back to node w. And then node w compares the results returned by the two nodes. Use $\sigma^*((u,v)_w)$ to express the result of node w comparing the outputs of the node u and node v. If the two results are the same then $\sigma^*((u,v)_w) = 0$, otherwise, $\sigma^*((u,v)_w) = 1$. If node w cannot communicate with node u and node v, then $\sigma^*((u,v)_w) = -1$. All the test results are called the comparative symptoms of this system, and are recorded as $\sigma^*((u,v)_w)$. The specific definition of the MM* diagnostic model is shown in Table 1.

TABLE 1. **MM* model.**

Comparative node w	Compared node u and v	Comparison result $\sigma^*((u,v)_w)$
Fault-free	Fault-free	0
Both fault-free	At least one is fault	1
Fault	Arbitrary state	0 or 1

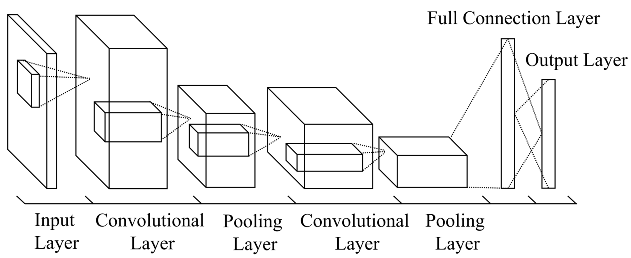


FIGURE 2. **Common convolutional neural network structure.**

In a t-diagnosable system, there is a correct and complete diagnostic algorithm, that is, all faulty nodes in the system can be identified completely and correctly. A system with n nodes is t-diagnosable if and only if the number of faulty nodes t is less than or equal to $(n-1)/2$, and each node can be tested by at least t other nodes. In a t-diagnosable system, the number of faulty nodes is always less than the number of fault-free nodes. The fault diagnosis of wireless sensor network in this paper is based on the premise of t-diagnosable system.

III. IMPROVED CONVOLUTIONAL NEURAL NETWORK BASED ON FIREWORKS ALGORITHM

A. CONVOLUTIONAL NEURAL NETWORK

Convolutional neural networks are an important model in deep learning, and it has now become a part of the era of computer vision. Convolutional neural networks, like ordinary full-path neural networks, are composed of neurons formed by weights and biases. The convolutional neural network reduces the number of parameters in the network using partial connection of neurons and does not cause loss of performance. Convolutional neural networks have excellent performance in the image field. The basic structure of convolutional neural networks includes local perception, shared weights, and pooling operations. Figure 2 is a diagram of a common convolutional neural network structure.

1) CONVOLUTIONAL LAYER

When the size of the convolution kernel is i, the relationship between the activation value a' of each node of the next layer and the activation value a of the node of the previous layer is as shown in Eq. (1). f is the activation function. w and b are weights and biases, respectively.

$$a'_{j,k} = f\left(\sum_{l=1}^{i}\sum_{m=1}^{i} w_{l,m}a_{j+l,k+m} + b\right) \tag{1}$$

Each feature map will have a feature map. In general, the convolution layer will have multiple feature maps, that is, corresponding to multiple convolution layer filters.

2) POOLING LAYER

In the convolutional neural network, the pooling layer is also a very common structural unit. The pooling layer reduces the data size by compressing the data. It is usually connected behind the convolutional layer. The principle is similar to the compressed image resolution. There is not much loss of image features. For example, the value of the next layer node in the average pooling layer corresponds to the average of four adjacent nodes in the upper layer of the hidden layer. When the pooling layer size is m, the average pooling layer can be expressed by Eq. (2).

$$y_{r,c} = \frac{\sum_{p=1}^{m}\sum_{q=1}^{m} x_{r*m+p,c*m+q}}{m^2} \tag{2}$$

where $y_{r,c}$ represents the output value at the (r, c) position, and the activation value at the (j, k) position of the previous layer is $x_{j,k}$. In addition to the average pooling layer, there is also a maximum pooling layer etc. The largest pooling layer is the maximum value of several adjacent nodes in the upper layer.

3) FULL CONNECTION LAYER AND CLASSIFIER

After a series of feature processing, the data enters the fully connected layer and is then classified by the classifier. In the convolutional neural network, the structure of the fully connected layer and the classifier is the same as that of the traditional neural network. In this paper, the multi-layer full connection plus the softmax layer is used as the classifier to obtain the final output. For example, if there are k sensors for fault detection, the network ends up with k probability outputs. The calculation of the fully connected layer is as shown in Eq. (3), where w is the weight, b is the offset. a' represents the activation value of the next layer. a represents the activation value of the previous layer.

$$a'_j = f\left(\sum_{l=1}^{i} w_l a_l + b\right) \tag{3}$$

The k activation values $y_1, y_2, \cdots, y_n$ are output after the multi-layer full connection, and the final classification result after the softmax regression processing is as shown in Eq. (4).

$$softmax\,(y)_i = \frac{e^{yi}}{\sum_{j=1}^{n} e^{yi}} \tag{4}$$

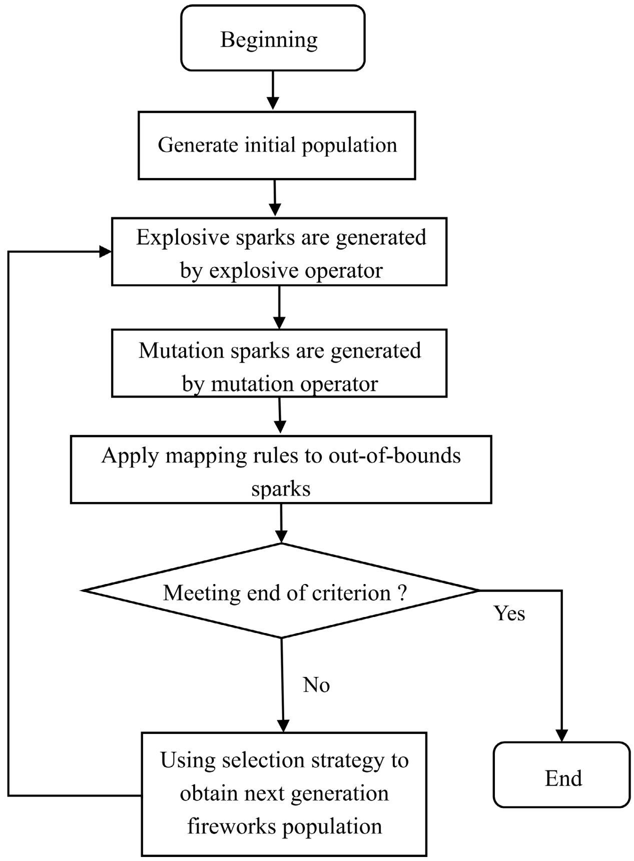


FIGURE 3. Flowchart of fireworks algorithm.

B. TRADITIONAL FIREWORKS ALGORITHM

The fireworks algorithm is an algorithm that simulates the process of a fireworks explosion. The fireworks explosion will generate a lot of sparks and spread in the local space. The sparks can continue to explode as new fireworks, gradually making the fireworks (sparks) fill the whole space. The basic principle of the fireworks algorithm is: the smaller the explosion radius of the better fireworks, the more sparks are generated by the explosion, to enhance the local search ability of the algorithm. On the contrary, the larger the explosion radius of the worse fireworks, the fewer sparks are generated by the explosion, to enhanced the global search capability of the algorithm. The fireworks algorithm simulates the fireworks explosion process in real life. Figure 3 shows the flow chart of the fireworks algorithm [23].

The population is initialized, that is N fireworks are randomly generated in a specific solution space (N is a suitable value equationted by the problem scale), and each fireworks individual x_i represents a feasible solution in the solution space.

The blast radius A_i and the number of sparks S_i generated by the explosion of the fireworks x_i are calculated according to the following equation:

$$A_i = \hat{A} \cdot \frac{f\,(x_i) - Y_{best} + \varepsilon}{\sum_{i=1}^{N} (Y_{best} - f\,(x_i)) + \varepsilon} \tag{5}$$

$$S_i = \hat{S} \cdot \frac{Y_{worst} - f(x_i) + \varepsilon}{\sum_{i=1}^{N} (Y_{worst} - f(x_i)) + \varepsilon} \tag{6}$$

Among them $\hat{A}$ and $\hat{S}$ are constants, which are used to limit the maximum explosion radius and the maximum number of sparks generated by the explosion. Y_{best} represents the best value of the fitness function corresponding to all fireworks, and Y_{worst} represents the worst value of the fitness function corresponding to all fireworks. $f(x_i)$ is the fitness value of fireworks individual x_i. ε is a very small constant, avoiding the case where the denominator has zero in the above equation.

The way in which fireworks individual x_i produces sparks (displacement operation) is as follows:

$$\hat{x}_i^k = x_i^k + \Delta x \tag{7}$$

where $\hat{x}_i^k$ is the position of the i explosion spark in the k dimension. x_i^k is the position of the i fireworks x_i in the k dimension. $\Delta x = A_i \times rand(-1, 1)$. $k = 1, 2, \cdots, D$, the dimension representing the problem to be optimized.

After the N fireworks explosions are completed, to increase the diversity, $\hat{m}$ Gaussian sparks are added, and each spark is calculated according to Eq. (8).

$$\tilde{x}_i^k = x_i^k \times Gaussian(1, 1) \tag{8}$$

where $\tilde{x}_i^k$ is the position of the i mutation spark in the k dimension. x_i^k is the position of the i fireworks in the k dimension. $Gaussian(1, 1)$ is the random number whose mean and variance are both 1.

To prevent the newly generated two spark particles from exceeding the search range, the fireworks algorithm uses the mapping rule of the modulo operation to pull the sparks outside the feasible range back to the feasible range. If the spark particle x_i is outside the feasible range, it is calculated as follows:

$$x_i^k = x_{min}^k + \left| x_i^k \right| \% \left(x_{max}^k - x_{min}^k \right) \tag{9}$$

where x_i^k is the position of the i fireworks x_i in the k dimension. x_{max}^k and x_{min}^k are the upper and lower search boundaries of the dimension, respectively. % is the modulo operation.

Apply the selection strategy to get the next generation of fireworks groups, that is, select N fireworks individuals from fireworks, explosion sparks, and mutation sparks to form the next generation fireworks population. The selection strategy is as follows: select the fireworks with the best fitness value directly for the next generation fireworks group, and the remaining $N-1$ fireworks individuals are selected according to the following probability:

$$R(x_i) = \sum_{j \in k} d(x_i - x_j) \tag{10}$$

$$p(x_i) = \frac{R(x_i)}{\sum_{j \in k} R(x_j)} \tag{11}$$

where $R(x_i)$ is the sum of the distances between the fireworks individual x_i and other individuals. According to the above equation, the probability that the fireworks (sparks) are selected is inversely proportional to the concentration, which is beneficial to the diversity of the population.

C. IMPROVEMENT STRATEGY OF FIREWORKS ALGORITHM

1) INITIAL POPULATION

The objects optimized by the firework algorithm are the weights and thresholds randomly generated by the neural network. For a given neural network structure, the connected weights and thresholds are directly arranged to form a firework individual. If a neural network has a three-layer structure, the weights and thresholds from the input layer to the hidden layer are $[W_1, W_2, \ldots, W_{n1}]$ and $[\theta_1, \theta_2, \ldots, \theta_{n2}]$, respectively, and the weights and thresholds from the hidden layer to the output layer are $[V_1, V_2, \ldots, V_{n2}]$ and $[r_1, r_2, \ldots, r_{n3}]$, respectively. Among them, W_i is a weight vector converted from the input layer to the hidden layer into a n_2-dimensional row vector, and V_i is a weight vector converted from the hidden layer to the output layer into an n_3-dimensional row vector. An individual firework in the population can be represented as $[W_1, W_2, \ldots, W_{n1}, \theta_1, \theta_2, \ldots, \theta_{n2}, V_1, V_2, \ldots, V_{n2}, r_1, r_2, \ldots, r_{n3}]$.

2) IMPROVEMENT OF EXPLOSION RADIUS

The main intention of Eq. (5) is that the explosion radius of the fireworks is inversely proportional to the fitness function value, which can ensure that the good fireworks enhance the local search. However, for the best fireworks, the value of the explosion radius calculated by putting into Eq. (5) is almost 0, which is obviously not following the original design intention of the algorithm. According to Eq. (6), the best fireworks produce the most sparks, which means that the best fireworks produce a lot of sparks, but did not conduct any search, increasing the amount of calculation.

In response to this problem, we need to introduce a lower bound of explosion radius $A_{min}(t)$ (which represents the minimum explosion radius of the first iteration) to limit the explosion radius, namely:

$$A_i = \begin{cases} A_{min}(t), & A_i < A_{min}(t) \\ A_i, & others \end{cases} \tag{12}$$

For $A_{min}(t)$, two schemes of linear decline and nonlinear decline are respectively presented in [24].

$$A_{min}(t) = A_{init} - \frac{A_{init} - A_{final}}{evals_{max}} * t \tag{13}$$

$$A_{min}(t) = A_{init} - \frac{A_{init} - A_{final}}{evals_{max}} \sqrt{(2 * evals_{max} - t)\, t} \tag{14}$$

where A_{init} represents the initial blast radius of the algorithm. A_{final} represents the blast radius at the end of the algorithm. $evals_{max}$ represents the maximum number of iterations.

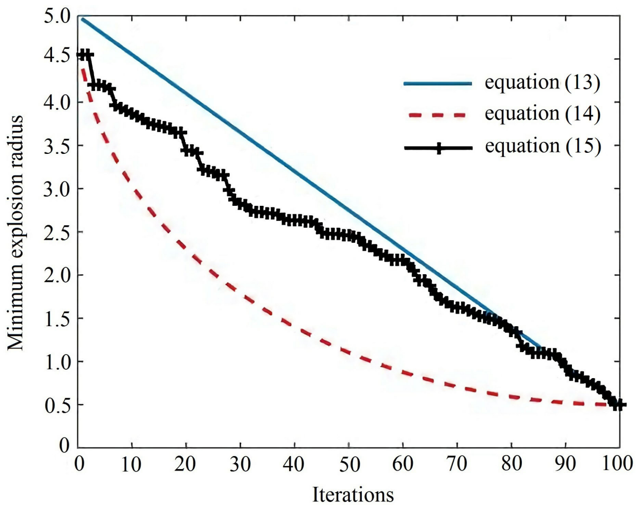


FIGURE 4. **The minimum radius varies with the number of iterations.**

t represents the number of iterations of the current algorithm evolution.

Similar improvements have been given in [25]–[27], but all introduce more parameters that require manual settings. To simplify the parameter setting and achieve a similar effect, this paper improves the explosion radius as follows:

$$A_{min}(t) = A_t + \frac{A_t * t}{100} \tag{15}$$

where $A_{min}(t)$ represents the minimum blast radius of the current iteration.

Figure 4 shows the change diagram of the minimum radius proposed by [24] and this paper, where $A_{init} = 5$, $A_{final} = 0.5$, $evals_{max} = 100$.

Obviously, the three minimum radius calculation methods can achieve similar results. It can be seen from Figure 4 that, with the increase of the iteration number t of algorithm evolution, the minimum explosion radius $A_{min}(t)$ decreases gradually. This makes the algorithm at the beginning of the search radius is large, focus on a global search. In the later stage, when the optimal value is nearly found, the explosion radius is reduced, which is conducive to fine local search. Compared with [25]–[27], Eq. (15) does not introduce parameters that need to be manually set and maintains the simplicity of the fireworks algorithm.

3) IMPROVEMENT IN DISPLACEMENT OPERATION

In the traditional fireworks algorithm, when an individual fireworks explosion produces an explosion spark, the offset occurring in each dimension is the same, which greatly reduces the diversity of the explosion spark population. Aiming at this defect, this paper introduces a chaotic sequence in the process of individual fireworks exploding to produce explosive sparks, which can realize the displacement operation with different magnitudes of offset on each dimension. The implementation process is shown in Eq. (16).

$$\tilde{x}_i^k = x_i^k + (3g - 1) A_i \tag{16}$$

where g is an element in a set of chaotic sequences between [0, 1].

4) IMPROVEMENT IN MUTATION OPERATOR

To allow the current fireworks individual to better share relevant information with the optimal fireworks individual, the genetic algorithm's mutation idea is introduced. For the current fireworks individual x_i, generate a random number between [0, 1]. If this random number is less than the set mutation probability, then the individual fireworks mutation operation, the equation is as follows:

$$\tilde{x}_i^k = x_{best}^k + h_i \left(x_{best}^k - x_i^k \right) \; if \; p_i \le p_m \tag{17}$$

where x_{best}^k is the position of the optimal fireworks individual x_{best} in the current population in the k dimension. h_i is the random number between $(-1, 1)$. p_i is the random number between [0, 1]. p_m is a pre-set mutation probability.

5) IMPROVEMENTS IN MAPPING RULES

When a fireworks (spark) individual exceeds the feasible domain, it can be mapped to a new location within the feasible domain by the following mapping rules:

$$x_i^k = x_{min}^k + U(0,1) \left(x_{max}^k - x_{min}^k \right) \tag{18}$$

where $U(0,1)$ is a random number uniformly distributed over [0, 1].

Theotem 1: all fireworks beyond the feasible region can be mapped to the feasible region through Eq. (18).

Proof: from the definition of MM*, we know that $x_{min}^k = -1$, $x_{max}^k = 1$, so $x_{max}^k - x_{min}^k = 2$.The value range of $U(0,1)$ is [0, 1], so the range of $U(0,1)\left(x_{max}^k - x_{min}^k\right)$ is [0, 2]. While $x_i^k = x_{min}^k + U(0,1)\left(x_{max}^k - x_{min}^k\right)$, it can be inferred that the value range of x_i^k is $[-1, 1]$.

So Theotem 1 is proved.

6) IMPROVEMENTS IN SELECTION STRATEGY

In the basic fireworks algorithm, the selection strategy is measured by distance. However, in this way of selection, the Euclidean distance between any two points needs to be calculated during each generation of population construction, which will result in a large time consumption of the basic fireworks algorithm. This paper adopts a semi-reserved elitist and semi-random selection strategy, that is, the former $N/2$ fireworks individuals with good fitness values are retained, and the remaining $N/2$ fireworks individuals adopt a random selection strategy. Form these N fireworks individuals into a new population set and enter the next iteration.

D. SOLVING THE INITIAL WEIGHT AND BIAS OF THE CONVOLUTIONAL NEURAL NETWORKS USING FIREWORKS ALGORITHM

The basic framework of the fireworks algorithm optimization convolutional neural network learning method proposed in this paper is shown in Figure 5. The main problem of the

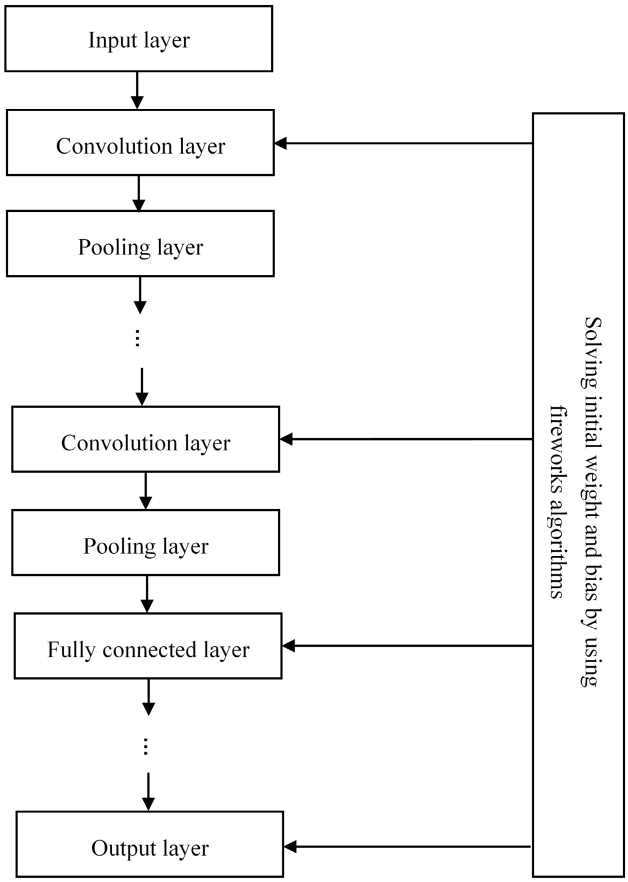


FIGURE 5. Algorithm framework of the convolutional neural network combined with fireworks algorithm.

traditional the convolutional neural network is that the learning performance is greatly affected by the initial weight and offset settings of each layer. To solve this problem, the initial weight and bias of each layer are optimized and solved in the training process of convolutional neural network in this paper. The specific algorithm is described as follows:

Algorithm 1 :Solving Initial Weight and Bias With Fireworks Algorithms

Input: W_i, θ_i, V_i, r_i
Output: x_{best}

```
for (int i = 1; i <= N; i++)
    Initialization of convolutional neural networks.
    Training convolutional neural networks by the steepest descent algorithm.
    Save the weight and bias of this training.
end for
The weight and bias combination of each group is coded into fireworks individuals, and the population of the fireworks algorithm is initialized.
Do {
    Calculate the fitness value of individual fireworks.
    Generating explosive sparks.
    Generating mutation sparks.
    Cross-border detection and mapping.
    Select the next generation of fireworks population.
}
While (The fitness value meets the termination condition or the number of iterations reaches the upper limit);
Decoding the optimal fireworks individual to obtain the initial weight and offset of the convolutional neural network.
```

Among them, the specific calculation steps of the fitness value of individual fireworks are as follows:

Step 1: Decode the individual fireworks to obtain a set of initial weights and offsets.

Step 2: The initial weights and biases of the group are used as the weights and biases of the corresponding layers of the convolutional neural network.

Step 3: Calculate the diagnostic accuracy of the convolutional neural network after training, and use it as the fitness value of the corresponding fireworks individual.

IV. FAULT DIAGNOSIS OF WSN BASED ON FWA-CNNFD

A. EXPERIMENTAL DESIGN

The algorithm design platform of this paper is Python, and the simulation experiment is carried out on a computer with 8.00GB of memory and Intel (R) Core (TM) i5-7400 3.00GHz.

According to the MM* model, we first set the number of sensor nodes as 300 and randomly generated network topology diagrams of different scales. Then, according to the t- diagnosable (the number of fault nodes in the network should not exceed half of the total number of nodes), fault nodes are randomly generated. Thus, the corresponding system test symptom S and fault mode F are obtained. That is the system test report. The specific implementation method is: first, initialize an all-zero array **A** of length 300. According to t-diagnosable, a fault node is randomly generated, that is, some elements are assigned a value of 1 (the number of 1 is less than the number of 0). This array represents 300 sensor nodes. If $\mathbf{A}[i]$=1, it means that the sensor node i is fault, otherwise, if $\mathbf{A}[i]$=0, it means that the sensor node i is fault-free. Then, randomly generate a 300×300 matrix **B** consisting of 0 and 1. If $\mathbf{B}[i][j]$=1, it means that there is a communication link between nodes i and j, otherwise, if $\mathbf{B}[i][j] = 0$, it means that there is no communication link between nodes i and j. The corresponding values are obtained through the values of the array **A** and matrix **B** and the method described in Table 1, and then a 300×300 diagnostic matrix of 300 nodes can be obtained. The index of each matrix represents the index of the array **A**, that is, the serial number of the node. As long as there is no communication link between a node corresponding to an index and a comparison node (the corresponding value of the **B** matrix is 0), the corresponding value of the diagnostic matrix is -1. Finally, a $300 \times 300 \times 300$ three-dimensional matrix **C** is obtained by stacking the diagnosis matrix of each node. Among them, **C** is the test symptom of the system, and **A** is the fault mode. In the

TABLE 2. The setting of relevant parameters of fireworks algorithm.

parameter name	Parameter Description	Parameter value
N	Fireworks population size	20
$\hat{S}$	Maximum spark number of fireworks	35
$\hat{A}$	Maximum explosion radius of fireworks	6
P_m	Mutation probability	0.3
x^k_{min}	Fireworks position lower bound	-1
x^k_{max}	Fireworks position upper bound	1
$goal$	Fitness value (termination condition)	0.9
$evals_{max}$	The maximum number of iterations	100

TABLE 3. The setting of relevant parameters of neural network training.

parameter name	Parameter Description	Parameter value
lr	Network learning rate	0.01
e	Number of iterations (parameter initialization phase)	50
epochs	Training maximum number of iterations (training phase)	1000

system test report, 80% was randomly selected as the training sample, and the remaining 20% was used as the test sample. Because the test symptoms and failure modes generated under the MM* model are both -1, 0, or 1, the data itself has been normalized, so no further normalization of the data is required in the subsequent diagnostic experiments.

The CNN network uses two convolutional layers (the convolution kernel size is set to 5, and the number is 16 and 32 respectively), two pooling layers (max-pooling of size 2), two fully connected layers, and a Softmax classifier. Adopt SGD (Stochastic Gradient Descent) optimization algorithm.

The performance evaluation index uses diagnostic AR (Accuracy Rate) and FAR (False Alarm Rate). In a WSN $G(U, E)$, the diagnostic accuracy and diagnostic false alarm rate are defined as:

$$AR = \frac{U_{TF}}{U_F} \times 100\% \tag{19}$$

$$FAR = \frac{U_{NF}}{U_{FF}} \times 100\% \tag{20}$$

In among U_{FF} represents a set of fault-free sensor nodes. U_F represents a set of faulty sensor nodes. U_{TF} represents a set of nodes that accurately diagnose the fault. U_{NF} represents a set of fault-free sensor nodes diagnosed as faulty nodes.

The fireworks algorithm and convolution neural network algorithm described in this paper need to configure some parameters. The relevant parameters of the fireworks algorithm are shown in Table 2. The relevant parameters of the convolution neural network are shown in Table 3.

B. ALGORITHM COMPARISON

In order to compare the performance of this method and other classical fault diagnosis methods for WSN, under the condition of the selection of the same training parameter, using the same test report (data sets), and then respectively

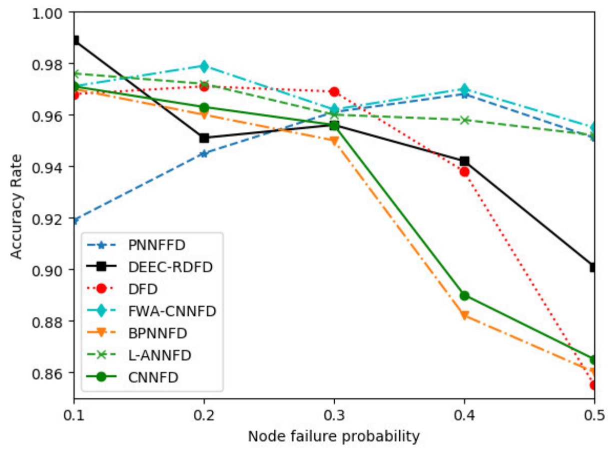


FIGURE 6. Comparison of diagnostic accuracy of seven algorithms at different node failure rates.

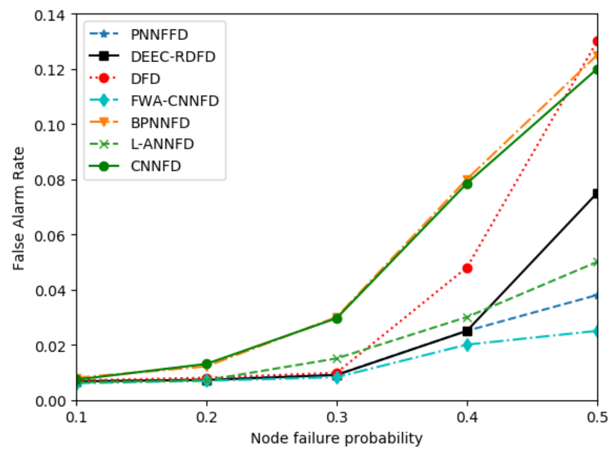


FIGURE 7. Comparison of diagnostic false alarm rates for seven algorithms at different node failure rates.

compare the diagnosis accuracy rate and false alarm rate of DFD algorithm [7], DEEC-RDFD algorithm [8], PNNFD algorithm [14], BPNNFD algorithm [16], L-ANNFD algorithm [17], CNNFD algorithm and FWA-CNNFD algorithm. Among them, the unique parameters of each algorithm are set according to the corresponding literature. In order to ensure the stability of the results and reduce their randomness, under the same conditions, this paper repeated 100 experiments, taking the average value of diagnostic accuracy and diagnostic false alarm rate. The results are shown in Figure 6 and Figure 7.

It can be seen from Figure 6 that with the increase of node failure rate, the diagnostic accuracy of the DFD algorithm, DEEC-RDFD algorithm, BPNNFD algorithm, CNNFD algorithm, and L-ANNFD algorithm show different degrees of decrease. The diagnostic accuracy of FWA-CNNFD algorithm is relatively stable. Although the diagnostic accuracy of the PNNFD algorithm keeps improving, it is always lower than that of the FWA-CNNFD algorithm. Because the experiment is conducted under the premise of t-diagnosable system, the closer the node failure rate is to 50%, the lower the accuracy of the algorithm. In particular, when the node failure rate exceeds 40%, the accuracy of the DFD algorithm, DEEC-RDFD algorithm, CNNFD algorithm, and BPNNFD algorithm decreases sharply, while the accuracy of the FWA-CNNFD algorithm still exceeds above 95%.

As shown in Figure 7, when the node failure rate increases, the diagnostic false alarm rate of the DEEC-RDFD algorithm, DFD algorithm, CNNFD algorithm, and BPNNFD algorithm rise sharply, while the false alarm rate of FWA-CNNFD algorithm increases the least among the seven algorithms. When the node failure rate is lower than 30%, the false alarm rate of the DFD algorithm, DEEC-RDFD algorithm, PNNFD algorithm, and FWA-CNNFD algorithm are less than 1%. However, when the node failure rate is as high as 50%, the false alarm rate of the FWA-CNNFD algorithm is only 2.5%, which is lower than the DFD algorithm. 10.5 percentage points.

The time complexity of the FWA-CNNFD algorithm is $O\left(N^2\right)$. Combined with Figure 6 and Figure 7, we can see that the wireless sensor fault diagnosis algorithm based on the fireworks algorithm and convolution neural network is more effective.

V. CONCLUSION

Using the efficient search ability of global and local optimal solutions of fireworks algorithm, and the local perception, parameter sharing, multi-core and multi-layer convolution of convolution neural networks, this paper proposes a fault diagnosis method for WSN nodes which combines fireworks algorithm and convolution neural network. Firework algorithm is a swarm intelligence algorithm with global search capability and local search capability self-adjusting mechanism. By combining the fireworks algorithm and the convolutional neural network, it can not only overcome the problems of the slow speed of the general convolutional neural network algorithm and easy to fall into the local minimum value, but also have the advantages of the convolutional neural network and the firework algorithm itself. According to the characteristics of WSN fault diagnosis based on MM* model, the explosion radius, mutation operator, mapping rule and selection strategy of fireworks algorithm are improved respectively. Then, the improved fireworks algorithm is used to generate the initial weight and bias required for convolutional neural network training, to avoid the problem that the performance of the steepest descent algorithm is greatly affected by the initial weight and bias setting when training the convolutional neural network. The simulation results show that the proposed fault diagnosis algorithm has higher diagnostic accuracy and a lower false alarm rate. It can effectively detect faulty nodes in WSN, and can well solve the problem of node fault in the network.

In this paper, the algorithm we designed has adopted some improved methods f to overcome the shortcomings of current research, which effectively reduces the error rate of node fault diagnosis. However, due to limited conditions, we only carried out experiments in the simulation environment and cannot simulate the influence of complex environmental conditions on sensor nodes in reality. The execution results of the diagnostic algorithm in the actual environment will also be affected by the real environment, producing results that are not in line with expectations. In this regard, we can measure the effectiveness of the algorithm in the actual environment in the future, to obtain more real and effective data.

REFERENCES

[1] F. P. Preparata, G. Metze, and R. T. Chien, "On the connection assignment problem of diagnosable systems," *IEEE Trans. Electron. Comput.*, vol. EC-16, no. 6, pp. 848–854, Dec. 1967.

[2] F. Barsi, F. Grandoni, and P. Maestrini, "A theory of diagnosability of digital systems," *IEEE Trans. Comput.*, vol. C-25, no. 6, pp. 585–593, Jun. 1976.

[3] K.-Y. Chwa and S. L. Hakimi, "Schemes for fault-tolerant computing: A comparison of modularly redundant and t-diagnosable systems," *Inf. Control*, vol. 49, no. 3, pp. 212–238, Jun. 1981.

[4] M. Malek, "A comparison connection assignment for diagnosis of multiprocessor systems," in *Proc. 7th Annu. Symp. Comput. Archit.*, La Baule-Escoublac, France, 1980, pp. 31–36.

[5] J. Maeng and M. Malek, "A comparison connection assignment for self-diagnosis of multiprocessor system," in *Proc. 11th Int. Symp. Fault-Tolerant Comput.*, Portland, OR, USA, 1981, pp. 173–175.

[6] A. Sengupta and A. T. Dahbura, "On self-diagnosable multiprocessor systems: Diagnosis by the comparison approach," *IEEE Trans. Comput.*, vol. 41, no. 11, pp. 1386–1396, Nov. 1992.

[7] A. De Paola, S. Gaglio, G. L. Re, F. Milazzo, and M. Ortolani, "Adaptive distributed outlier detection for WSNs," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 902–913, May 2015.

[8] Y. Zhang and X. Yuan, "Fault diagnosis in clustering WSN based on neighbor cooperation," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Yinchuan, China, May 2016, pp. 1803–1807.

[9] R. R. Swain and P. M. Khilar, "Soft fault diagnosis in wireless sensor networks using PSO based classification," in *Proc. IEEE Region 10 Conf. (TENCON)*, Penang, Malaysia, Nov. 2017, pp. 2456–2461.

[10] R. R. Swain, P. M. Khilar, and S. K. Bhoi, "Heterogeneous fault diagnosis for wireless sensor networks," *Ad Hoc Netw.*, vol. 69, pp. 15–37, Feb. 2018.

[11] Y. Gao, F. Xiao, J. Liu, and R. Wang, "Distributed soft fault detection for interval type-2 fuzzy-model-based stochastic systems with wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 334–347, Jan. 2019.

[12] Y. Cheng, Q. Liu, J. Wang, S. Wan, and T. Umer, "Distributed fault detection for wireless sensor networks based on support vector regression," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–8, Oct. 2018.

[13] W. He, P.-L. Qiao, Z.-J. Zhou, G.-Y. Hu, Z.-C. Feng, and H. Wei, "A new belief-rule-based method for fault diagnosis of wireless sensor network," *IEEE Access*, vol. 6, pp. 9404–9419, 2018.

[14] M. Mengying *et al.*, "Fault diagnosis algorithm of WSN based on precondition of neighbor nodes," *J. Comput. Appl.*, vol. 38, no. 8, pp. 2348–2352, 2018.

[15] X. Wu, H. Chen, Y. Wang, L. Shu, and G. Liu, "BP neural network based continuous objects distribution detection in WSNs," *Wireless Netw.*, vol. 22, no. 6, pp. 1917–1929, Aug. 2016.

[16] R. R. Swain and P. M. Khilar, "Composite fault diagnosis in wireless sensor networks using neural networks," *Wireless Pers. Commun.*, vol. 95, no. 3, pp. 2507–2548, Aug. 2017.

[17] R. R. Swain, P. M. Khilar, and T. Dash, "Neural network based automated detection of link failures in wireless sensor networks and extension to a study on the detection of disjoint nodes," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 2, pp. 593–610, Feb. 2019.

[18] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.

[19] Y. Chen, L. Li, X. Zhao, J. Xiao, Q. Wu, and Y. Tan, "Simplified hybrid fireworks algorithm," *Knowl.-Based Syst.*, vol. 173, pp. 128–139, Jun. 2019.

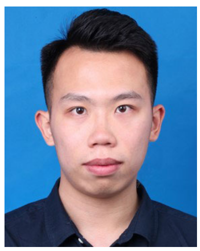
[20] Q. Lu, W. Gui, and M. Su, "A fireworks algorithm for the system-level fault diagnosis based on MM* model," *IEEE Access*, vol. 7, pp. 136975–136985, 2019.

[21] Z. Zhang, A. Mehmood, L. Shu, Z. Huo, Y. Zhang, and M. Mukherjee, "A survey on fault diagnosis in wireless sensor networks," *IEEE Access*, vol. 6, pp. 11349–11364, 2018.
[22] M. Jeyaselvi and C. Jayakumar, "Enhanced clone detection using channel response information in wireless sensor network," *J. Comput. Theor. Nanosci.*, vol. 14, no. 9, pp. 4174–4182, Sep. 2017.
[23] Y. Tan and Z. Yuanchun, "Fireworks algorithm for optimization," in *Proc. Int. Conf. Swarm Intell.*, Beijing, China, 2010, pp. 355–364.
[24] C. Yu, L. Kelley, S. Zheng, and Y. Tan, "Fireworks algorithm with differential mutation for solving the CEC 2014 competition problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Beijing, China, Jul. 2014, pp. 3238–3245.
[25] J. Li, S. Zheng, and Y. Tan, "Adaptive fireworks algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Beijing, China, Jul. 2014, pp. 3214–3221.
[26] S. Zheng, A. Janecek, J. Li, and Y. Tan, "Dynamic search in fireworks algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Beijing, China, Jul. 2014, pp. 3222–3229.
[27] B. Zhang, M.-X. Zhang, and Y.-J. Zheng, "A hybrid biogeography-based optimization and fireworks algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Beijing, China, Jul. 2014, pp. 3200–3206.

WEIXIA GUI is currently an Associate Professor with Guangxi University, Guangxi, China, where she is engaged in research and development of intelligent computing, networks, and parallel distributed computing.

QIAN LU is currently pursuing the master's degree with Guangxi University, Guangxi, China, where she is engaged in research and development of intelligent algorithms and parallel computing.

MEILI SU is currently pursuing the master's degree with Guangxi University, Guangxi, China, where she is engaged in research and development of intelligent algorithms and parallel computing.



FULAI PAN is currently pursuing the master's degree with Guangxi University, Guangxi, China, where he is engaged in research and development of intelligent algorithms and parallel computing.