Research Article

A Novel Fault Diagnosis Strategy for Heterogeneous Wireless Sensor Networks

Li Cao ^{1,2}, Yinggao Yue ^{1,3} and Yong Zhang ³

¹Oujiang College, Wenzhou University, Wenzhou 325035, China

²School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China

³Computer School, Hubei University of Arts and Science, Xiangyang 441053, China

Correspondence should be addressed to Yong Zhang; zhang.yong@hbuas.edu.cn

Received 27 December 2020; Accepted 6 July 2021; Published 6 August 2021

Academic Editor: Bin Gao

Copyright © 2021 Li Cao 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.

Fault diagnosis is a guarantee for the reliable operation of heterogeneous wireless sensor networks, and accurate fault prediction can effectively improve the reliability of wireless sensor networks. First, it summarizes the node fault classification and common fault diagnosis methods of heterogeneous wireless sensor networks. After that, taking advantage of the short learning time, fewer parameter settings, and good generalization ability of kernel extreme learning machine (KELM), the collected sample data of the sensor node hardware failure is introduced into the trained kernel extreme learning machine and realizes the fault identification of various hardware modules of the sensor node. Regarding the regularization coefficient C and the kernel parameter s in KELM as the model parameters, it will affect the accuracy of the fault diagnosis model of the kernel extreme learning machine. A method for the sensor nodes fault diagnosis of heterogeneous wireless sensor networks based on kernel extreme learning machine optimized by the improved artificial bee colony algorithm (IABC-KELM) is proposed. The proposed algorithm has stronger ability to solve regression fault diagnosis problems, better generalization performance, and faster calculation speed. The experimental results show that the proposed algorithm improves the accuracy of the hardware fault diagnosis of the sensor nodes and can be better applied to the node hardware fault diagnosis of heterogeneous wireless sensor networks.

1. Introduction

Heterogeneous wireless sensor networks (HWSNs) are a kind of distributed sensor network, which is composed of a large number of fixed or mobile wireless sensor nodes in the form of self-organization and multihop transmission [1, 2]. The monitoring data collected by the sensor nodes can be transmitted between multiple nodes in a hop-by-hop manner. With the expansion of the scale of sensor networks, practical applications are faced with harsh environments such as high humidity, high and low temperatures, corrosive gases, high-energy particle radiation, drastic changes in pressure, and strong electromagnetic noise interference [3–5]. At the same time, the sensor nodes during their life cycle are affected by factors such as the manufacturing process, actual environment, and node energy, which are also very prone to failure and affect the network structure. This also poses great challenges to the connectivity, reliability, security, and life-

time of HWSNs. This is the key problem that needs to be solved urgently in the widening of application fields of heterogeneous wireless sensor networks. The sensor nodes of heterogeneous wireless sensor networks are often deployed in some special areas, and it is often impossible for humans to directly maintain equipment [6, 7]. Therefore, it is urgent to apply practical fault diagnosis technology to the node fault diagnosis of heterogeneous wireless sensor networks.

Through the fault diagnosis of the sensor node, various abnormal conditions and fault states of the sensor node can be timely and accurately diagnosed [8]. At the same time, predictive measures are given to guide the reliable and stable operation of the sensor network to ensure the stability, reliability, and availability of the network [9, 10]. The failure loss of the sensor node hardware is reduced to a minimum, so that the sensor node can effectively play the maximum role. The hardware failure of sensor nodes is the basis and prerequisite for various failures of heterogeneous wireless sensor

networks, and it is also a problem that needs to be solved urgently in the diagnosis of heterogeneous WSNs, and its research has very important significance [11, 12].

1.1. Problem Statement and Motivate. Heterogeneous wireless sensor networks have made considerable progress in hardware design, data processing performance, and communication stability. At the same time, the market's demand for network reliability and sustainability is also increasing. From the perspective of the entire wireless sensor network, the network is required to have better environmental applicability. On the other hand, the sensor nodes in wireless sensor networks require better robustness. Due to the influence of many unavoidable factors and the use of WSNs in extremely complex and harsh environments, the probability of sensor node failure is much higher than other systems. A faulty sensor node will cause data errors or missing during the collection cycle, directly reducing the reliability of the collected data, and ultimately affecting the overall decision-making of the system.

In practical applications, because the monitoring area is generally large, the external environment where the node is located is harsh and complex, and various faults are often prone to occur. The reliability and work continuity requirements of monitoring network operations are gradually increasing. After a node fails, it is prone to node failure, unreliable, or distorted data collected, causing network crashes and other problems. The probability of hardware failure of sensor nodes in HWSNs is much higher and more serious than other problems. It is particularly important to study fault diagnosis methods of sensor nodes in heterogeneous wireless sensor networks.

1.2. Contribution. In this work, a node fault diagnosis method of the heterogeneous sensor network based on kernel extreme learning machine optimized by the artificial bee colony algorithm (ABC-KELM) is proposed. In comparison with the current general selection approaches, the main contributions of our work in this paper can be summarized as follows:

- (1) Characterize the issues of the node fault diagnosis for HWSNs and classify the current fault diagnosis of the fault diagnosis methods
- (2) Propose a node fault diagnosis method of the heterogeneous sensor network based on kernel extreme learning machine optimized by the improved artificial bee colony algorithm (IABC-KELM).
- (3) Evaluate the performance of the proposed algorithms by comparing them with the fault diagnosis methods of the KELM, PSO-KELM, and ABC-KELM algorithm

The remainder of this paper is organized as follows: Section 2 discusses the related work and classifies the current fault diagnosis of the fault diagnosis methods. Section 3 describes the basic principles of kernel extreme learning machine. Section 4 Describes the algorithm design idea of kernel extreme learning machine optimized by the artificial bee colony algorithm, and Section 5 designs the implementation fault diagnosis steps of iABC-KELM for HWSNs. Sec-

tion 6 provides the parameters and simulation results that validate the performance of the proposed algorithm. Section 7 concludes the paper.

2. Classification of Node Hardware Failure and Related Work

Generally speaking, a sensor node is composed of a power module, a control processor module, a sensor module, a storage module, and a communication module. From the perspective of the functional division of wireless sensor networks, it is mainly composed of terminal nodes, routing nodes, and sink nodes. The terminal node completes the collection of perception data, the routing node is responsible for the data transmission, and the sink node is responsible for data collection and uploading to the processing center. The terminal node is particularly important in the entire data collection process. The terminal node is mainly composed of five modular modules: power supply module, CPU control module, wireless communication module, sensing module, and storage module. The fault classification of the node hardware of the heterogeneous wireless sensor network is shown in Figure 1. The node hardware failures of wireless sensor network mainly include impact fault, short circuit fault, bias fault, offset fault, periodic interference fault, nonlinear dead time fault, open circuit fault, and drift fault. The common faults and causes of sensor components are shown in Table 1.

There are many reasons for the failure of WSNs' sensor nodes, and the effects of the failure are also different. Generally, most sensor nodes are active, and sensing nodes are distributed in harsh application environments. The nodes are powered by dry batteries, and it is inconvenient to replace the power supply or charge. When the node is working, the energy is gradually consumed. When it is lower than the normal working setting value, the node is unstable, the measurement is inaccurate, and the collected data has a large error, which affects the normal operation of the monitoring system. Common faults of the sensor nodes mainly include impact faults, short circuit faults, bias faults, battery exhaustion, periodic interference, nonlinear dead zone faults, open circuit faults, and temperature drift faults. The commonly used methods for fault diagnosis of the sensor nodes are shown in Figure 2.

At present, many experts and scholars have begun to conduct a lot of research on fault diagnosis of sensor nodes and have achieved some results. Shahriar et al. [13] studied the problem of recovering a batch of VNs affected by a substrate node failure. The combinatorial possibilities of alternate embeddings of the failed virtual nodes and links of the VNs make the task of finding the most efficient recovery both nontrivial and intractable. It is reported in [14] that the author proposed a data transmission scheme considering node failure for finishing validation of block data on blockchain, which firstly sets response threshold level to detect failure node, and then using greedy idea constructs communication tree to organize all nodes forwarding block data. Ghimire et al. [15] proposed a fault diagnosis method based on rough set theory for the problem of wireless sensor network node hardware fault diagnosis, which effectively

FIGURE 1: The sensor node hardware fault classification of HWSNs.

TABLE 1: Common faults and causes of sensor components.

Fault type	Cause of failure
Impact fault	Random interference in power supply and ground wire, electric spark discharge
Short circuit fault	Corrosion of bridges and short circuit connections caused by pollution
Bias fault	Bias current or bias voltage, etc.
Offset fault	The battery voltage cannot reach the minimum operating voltage of the node
Periodic interference fault	Power supply 50 Hz interference, etc.
Nonlinear dead time fault	The amplifier is saturated, contains nonlinear links, etc.
Open circuit fault	The signal line is broken, the chip pin is not connected, etc.
Drift fault	Temperature drift, etc.

diagnoses and locates each module of the sensor node with high diagnostic accuracy. Islam et al. [16] researched the fault diagnosis of sensors and proposed a wavelet packet-based fault diagnosis method, which can be effectively applied to sensor perception diagnosis applications. Gangsar et al. [17] proposed a heterogeneous wireless sensor network fault diagnosis method based on support vector machine to improve the fault diagnosis accuracy and network performance for the problem of low diagnosis accuracy of sensor node fault diagnosis. Cheng et al. [18] proposed an improved DFD node fault diagnosis method to greatly improve the accuracy of node fault diagnosis. Preeth et al. [19] proposed a sensor node fault diagnosis method based on artificial immune and fuzzy k -means. The problem was transformed into the problem of obtaining the optimal solution of the objective function, and the intelligent optimization algorithm was introduced to accurately realize node fault diagnosis.

The above methods partially improve the fault diagnosis accuracy of the node hardware or introduce intelligent optimization methods to improve the accuracy of the fault

diagnosis. However, these algorithms have more complex calculations, poor algorithm convergence speed, and sacrifice network energy to improve the accuracy of fault diagnosis. In this paper, by introducing the machine method, the kernel extreme learning machine has the characteristics of the short learning time, few parameter settings, and good generalization ability, and the collected fault sample data of the node hardware is introduced into the trained extreme learning machine. Regarding the regularization coefficient C and the kernel parameter s of the kernel extreme learning machine as the optimization target, these problems affect the accuracy of the fault diagnosis model of the nodes. Therefore, a node fault diagnosis method of the heterogeneous sensor network based on kernel extreme learning machine optimized by the artificial bee colony algorithm (ABC-KELM) is proposed in this paper, and the method realizes the identification of the hardware faults of the sensor nodes. The proposed algorithm is simple to implement, fast in network training, and high in fault diagnosis accuracy, which can effectively improve the reliability and stability of the network.

FIGURE 2: Commonly methods for fault diagnosis of HWSNs.

3. Kernel Extreme Learning Machine

Kernel extreme learning machine (KELM) is a new machine learning algorithm that combines kernel learning and extreme learning machine. In 2012, Huang et al. conducted indepth research on least squares support vector machines and found that kernel functions have inherent advantages in processing large-scale and complex data [20]. Later, Huang et al. introduced the kernel function into the ELM to form an extreme learning machine (KELM) with least square optimal solution. The algorithm data processing is simpler, can reduce the adjustment parameters of the algorithm, and improve the convergence speed of the algorithm [21]. The kernel extreme learning machine (KELM) introduces the kernel function into the extreme learning Machine.

It is a decision model with stronger robustness and better generalization performance than the conventional ELM model. The kernel extreme learning machine is composed of a data generator, a trainer, and a kernel learning machine [22]. The structure of KELM is shown in Figure 3. Among them, the data generator is used to generate a series of sample vectors x of unknown probability distribution functions, and the trainer can be used to distinguish sample categories or fit actual data models. The kernel extreme learning machine uses the data set $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ composed of several observed samples to construct an appropriate function to approximate the actual trainer [23].

Suppose there are N arbitrary samples (X_i, t_i) , where $X_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{in})^T \in R^n$ and $t_i = (t_{i1}, t_{i2}, t_{i3}, \dots, t_{im})^T \in R^m$, for a single-layer neural network with L hidden

FIGURE 3: The structure diagram of the kernel extreme learning machine.

neurons, the hidden layer output can be expressed as a matrix

$$H \times \beta = T. \quad (1)$$

In order to minimize the system training error, we need to consider minimizing empirical risk and minimizing structural risk. Therefore, here, we also minimize the error and output weights, which can be expressed by the following formula.

$$\min : L = \frac{1}{2} \|\beta\|^2 + \frac{1}{2} C \sum_{i=1}^N \|\xi_i\|^2, \quad (2)$$

$$st : h(x_i) \times \beta = t_i^T - \xi_i^T, i = 1, 2, \dots, N.$$

Wherein, the parameters are

$$\beta = \begin{bmatrix} \beta_1^T \\ \beta_2^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m} \quad (3)$$

and $\xi_i = [\xi_{i,1}, \xi_{i,2}, \dots, \xi_{i,m}]$, and it is the error between the network output corresponding to the training sample x_i and the true value. According to the KKT condition, the above solution formula is equivalent to

$$\min : L_{ELM} = \frac{1}{2} \|\beta\|^2 + \frac{1}{2} C \sum_{i=1}^N \|\xi_i\|^2 - \sum_{i=1}^N \sum_{j=1}^m \alpha_{ij} (h(x_i) \beta_j - t_{ij} + \xi_{ij}). \quad (4)$$

The parameter α_{ij} represents the Lagrangian operator of the i th sample, and the parameter C is a nonnegative constant. The corresponding optimization constraints are as follows:

$$\frac{\partial L_{ELM}}{\partial \beta_j} = 0 \longrightarrow \beta_j = \sum_{i=1}^N \alpha_{ij} (h(x_i)^T \longrightarrow \beta = H^T \times \alpha, \quad (5)$$

$$\frac{\partial L_{ELM}}{\partial \varepsilon_i} = 0 \longrightarrow \alpha_i = C \times \varepsilon_i, i = 1, 2, \dots, N, \quad (6)$$

$$\frac{\partial L_{ELM}}{\partial \alpha_i} = 0 \longrightarrow h(x_i) \beta - t_i^T + \varepsilon_i^T = 0, i = 1, 2, \dots, N. \quad (7)$$

Putting the formulas (5), (6), and (7) into formula (4), we can get

$$\beta = H^T \left(\frac{I}{C} + HH^T \right)^{-1} T. \quad (8)$$

Put formula (8) into the equation $f(x) = h(x) \times \beta$ to get the classification decision function:

$$f(x) = \text{sign} \left(h(x) H^T \left(\frac{I}{C} + HH^T \right)^{-1} T \right). \quad (9)$$

Usually, the training data N is much larger than the number of hidden nodes L of the extreme learning machine; so, we can get

$$\beta = CH^T \varepsilon, \quad (10)$$

$$\varepsilon = \frac{1}{C} (H^T)^+ \beta. \quad (11)$$

Putting formulas (10) and (11) into formula (8), we can get

$$H \times \beta - T + \frac{1}{C} (H^T)^+ \beta = 0, \quad (12)$$

$$H^T \left(H + \frac{1}{C} (H^T)^+ \right) = H^T T,$$

$$\beta = \left(\frac{1}{C} + HH^T \right)^{-1} H^T T.$$

The approximation function of the training function can be written as

$$f(x) = h(x) \left(\frac{I}{C} + HH^T \right)^{-1} H^T T. \quad (13)$$

According to the kernel theory, the activation function of the extreme learning machine is written in the form of a kernel function:

$$\Omega_{KELM} = H^T T : \Omega_{KELM} = h(x_i) h(x_j) = K(x_i, x_j). \quad (14)$$

The expression of the kernel extreme learning machine is

$$f(x) = h(x) H^T \left(\frac{I}{C} + HH^T \right)^{-1} T = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix} \left(\frac{I}{C} + HH^T \right)^{-1} T, \quad (15)$$

where the parameter H is the hidden layer matrix of the neural network, the parameter H^T is the generalized inverse matrix of H , the parameter T is the prediction target vector, and the parameter $1/C$ represents the diagonal element of the symmetric matrix plus the offset $1/C$. It can be seen from formula (15) that the kernel extreme learning machine is an optimized solution obtained by combining the kernel learning theory and the standard optimization methods. Due to relatively weak optimization constraints, the KELM algorithm has better generalization performance.

The characteristics of only one parameter (s) in the radial basis function (RBF) are more beneficial to optimization; so, the RBF radial basis function is used as the kernel function, namely,

$$K(x, y) = \exp \left(-\frac{\|x - y\|^2}{2s^2} \right). \quad (16)$$

Since the regularization factor C and the kernel parameter s need to be set in advance, the performance of the KELM method is greatly affected by these two parameters. The steps of the kernel extreme learning machine algorithm are as follows:

Step 1. Initialize the data sample N , $N = \{(x_i, t_i) | x_i \in R^n, t_i \in R^m, i = 1, 2, \dots, N\}$, and select the kernel function $K(u, v)$ of the kernel extreme learning machine and the regularization factor C .

Step 2. Calculate the hidden layer output matrix H of KELM.

Step 3. Calculate the output function of the KELM trainer.

Compared with the extreme learning machine algorithm, the kernel extreme learning machine has a stronger ability to solve regression prediction problems, better generalization performance, and faster calculation speed when it obtains better or similar prediction accuracy. In the kernel extreme learning machine algorithm, its own characteristics determine that the specific form of the feature mapping function in the hidden layer node does not need to be given, but the value of the output function can be obtained only by knowing the form of the kernel function. At the same time, there is no need to set the number of hidden layer nodes when solving the output function value; so, there is no need to set the initial weight and bias values of the hidden layer.

However, in actual application of the KELM method, its performance is affected by important parameters in the model. Practice has proved that the classification accuracy of KELM is affected by the regularization coefficient C and the setting of the kernel function parameter s , and the classification accuracy is easily trapped in a local minimum. Therefore, the parameters of KELM are improved by the artificial bee colony optimization algorithm to obtain the optimal hardware fault diagnosis model of the sensor nodes.

4. Kernel Extreme Learning Machine Optimized by the Artificial Bee Colony Algorithm (ABC-KELM)

The main idea of the algorithm is to use the better search ability of the artificial bee colony algorithm to optimize the regularization factor C and the kernel parameter s of KELM and find the optimal parameters or approximate optimal parameters and a more compact network structure. While ensuring to maximize the classification accuracy of the KELM model, minimize the output weight norm as much as possible to further improve the generalization ability and classification results of the KELM model.

4.1. Artificial Bee Colony Algorithm. The artificial bee colony algorithm (ABC) imitates the advanced swarm intelligence behavior exhibited by bee colony foraging behavior, searching for the best nectar source nearby through two mechanisms of division of labor and role conversion between different bees [24]. The bee colony carries out different activities according to their division of labor and realizes the information sharing and interaction of the entire bee colony through the form of swing dance. This method enables the bee colony algorithm to converge quickly, find the global optimal solution quickly, and is robust when solving combinatorial optimization problems. At the same time, the ABC algorithm has the advantages of fewer parameters and simple implementation and has been widely used in many fields such as polynomial function optimization, neural network parameter optimization, and industrial system design optimization [25]. The bees in the algorithm can be divided into three types: the employed bees, the observation bees, and the scout bees. The location of the food source represents a set of feasible solutions to the problem to be optimized, and the amount of nectar represents the fitness value of the feasible solution. The number of the employed bees and food sources is equal, and the location of each food source corresponds to one employed bee [26].

Assuming that the number of food sources is SN , the fitness value of the i -th food source is fit_i .

The employed bee searches for food sources according to formula (17). If a new food source v_{ij} is found in the neighborhood of the existing food source u_{ij} , and the fitness value of v_{ij} is greater than the fitness value of u_{ij} , the employed bee will replace u_{ij} with v_{ij} ; otherwise, keep u_{ij} .

$$v_{ij} = u_{ij} + \varphi_{ij}(u_{ij} - u_{kj}). \quad (17)$$

In the formula, the parameter φ_{ij} is a random value between $[-1, 1]$, the parameter k is a random number between $[1, SN]$, $k \neq i$, $1 \leq i \leq SN$, $1 \leq j \leq D$, and the parameter D is the dimension of the feasible solution.

When all the employed bees have completed the search, they will share the food source information with the observation bees in the hive. The observing bee will select the i th ($i = 1, 2, 3, \dots, SN$) food source according to the roulette method; that is, the greater the probability p_i corresponding

to the food source, the greater the probability of being selected. The calculation formula of the parameter p_i is

$$p_i = \frac{\text{fit}_i}{\sum_{k=1}^{SN} \text{fit}_k}. \quad (18)$$

Then, the observation bee searches for the new food sources in the selected food source neighborhood, calculates and compares the fitness values of the new and old food sources, and retains the food sources with a large fitness value.

When the food source u_{ij} reaches the limit cycle after the number of iterations, the position information is still not updated. Correspondingly, the observing bee abandons the food source and becomes a detective bee and searches for a new food source according to formula (19).

$$u_{ij} = u_{\min,j} + \text{rand}(0, 1)(u_{\max,j} - u_{\min,j}). \quad (19)$$

In formula (19), the parameters $u_{\min,j}$ and $u_{\max,j}$ are the upper and lower limits of the j th element in the feasible solution.

The ABC algorithm has the advantages of the simple calculation process, fewer parameters to be set, strong global search capability, and good robustness. The ABC algorithm is mainly used to find the optimal ELM input layer weight and hidden layer threshold [27].

The ABC algorithm has many advantages such as group intelligence, few parameters and easy operation, superior convergence speed and convergence accuracy, good robust performance, and good expansion performance, which makes the ABC algorithm widely concerned and tried to be applied to engineering optimization problems. However, when the basic ABC algorithm is applied to solve the high-dimensional complex optimization problems, it is similar to other swarm intelligence optimization algorithms and has obvious defects such as easy to fall into local optimal solutions and poor convergence accuracy [28]. Therefore, some researchers have improved the ABC algorithm. The above improvement strategies have accelerated the convergence speed and accuracy of the algorithm to a certain extent. However, there is still the possibility of premature convergence when facing the complex high-dimensional optimization problem to be solved and the possibility of falling into the local optimal solution. If the ABC algorithm is premature,

the Cauchy mutation operation is used; otherwise, the algorithm iteration continues.

When the artificial bee colony algorithm falls into the local optimal solution, because the standard ABC algorithm lacks the ability to jump out of the local optimal solution, this paper introduces a disturbance mechanism into the algorithm at this time to improve the ability of the ABC algorithm to jump out of the local optimal solution. The specific perturbation implementation method is as follows: first, perform Cauchy mutation perturbation on the n optimal individuals copied with a certain probability p to increase the diversity of the population. Then, the optimal individual after n mutations is optimized twice, giving the algorithm the ability to jump out of the local optimal solution. The Cauchy mutation operation is as follows:

$$X_i = X_{\text{best}} + \beta \times \text{Cauchy}(0, 1), i = 1, 2, 3, \dots, n. \quad (20)$$

In formula (20), the parameter X_{best} is the current optimal individual position in the colony, and the parameter β is the disturbance amplitude adjustment parameter, which is 0.5. $\text{Cauchy}(0,1)$ is the random number generator that satisfies the Cauchy distribution.

4.2. Kernel Extreme Learning Machine Optimized by the Artificial Bee Colony Algorithm. The improve kernel extreme learning machine optimized by the artificial bee colony algorithm is proposed in this paper, which has the following advantages in fault diagnosis of HWSNs. First, the ABC algorithm based on Cauchy mutation is a global search algorithm that can avoid local optima. Secondly, compared with traditional neural networks that need to optimize the input layer weights, hidden layer thresholds, and output weights, this method only needs to optimize the regularization factor C and the kernel parameter s of KELM. Then, the output weight is calculated, which improves the training speed and has better generalization ability.

The specific steps for optimization are as follows:

- (1) Randomly generate SN food sources and initialize the maximum number of iterations MCN , the maximum lag parameter limit, and the colony size. The location of each food source θ_i ($i = 1, 2, 3, \dots, SN$) is a set of input layer weights and hidden layer thresholds, and the calculation formula of food source θ_i is

$$\theta_i = [W_{11}, W_{12}, \dots, W_{1M}, W_{22}, \dots, W_{2M}, \dots, W_{n1}, W_{n2}, \dots, W_{nM}, b_1, b_2, \dots, b_M]. \quad (21)$$

In formula (21), the parameter W_{ij} is initialized to a random value between $[-1, 1]$, and the parameter b_i is initialized to a random value between $[0, 1]$. The parameters n and M are the number of the input layer and the hidden layer nodes, respectively.

- (2) The calculation formula of the fitness function is

$$\text{fit}_i = \frac{1}{1 + e_i}. \quad (22)$$

The parameters in formula (22) are the input layer weights of KELM corresponding to the i th food source and the mean squared error (MSE) of the hidden layer threshold training. The calculation formula of the mean squared error e_i is

$$e_i = \frac{1}{N} \sum_{i=1}^N (Y'_i - Y_i)^2. \quad (23)$$

In formula (23), the parameters Y'_i and Y_i are the actual output value and the expected output value, and the parameter N is the number of samples.

The employed bees search for a new food source V_{ij} in the neighborhood of the existing food source U_{ij} according to formula (17), calculates and compares the fitness value between the new and old food source, and retains the food source with a large fitness value. The dimension D of the feasible solution is

$$D = n \times M + M. \quad (24)$$

- (3) After all the employed bees have selected their food sources, the observation bee selects the search object according to the probability p_i of the i th ($i = 1, 2, 3, \dots, SN$) food source. Then, search for new food sources in the neighborhood of the i th food source according to formula (17) and keep the food sources with large fitness values
- (4) A certain food source has not been updated after the limit iterations, and the bee corresponding to the food source searches for a new food source according to formula (19)
- (5) Repeat steps (2) to (4) until the number of iterations reaches MCN, and the loop ends
- (6) The parameters corresponding to the food source with the largest fitness value in the cycle are used as the optimal input layer weight and hidden layer threshold of KELM

Decoding from the returned optimal solution can obtain the optimal input layer weight and hidden layer threshold. The flow chart of the artificial bee colony optimization extreme learning machine is shown in Figure 4.

The time complexity indirectly reflects the length of time the algorithm executes. In the ABC-KELM algorithm, it is assumed that the execution time required to initialize the parameters (under the condition that the population size is N , and the spatial dimension is n) is x_1 , and the time to generate a uniform distribution is x_2 . The time required to find the fitness value is $f(n)$, and then the time complexity of the initial stage of the ABC-KELM algorithm is as follows:

$$O(x_1 + N(nx_2 + f(n))) = O(n + f(n)). \quad (25)$$

Assuming that the execution time required for the iterative update of each dimension of the individual is the same, which is x_3 , the time for comparing the advantages and dis-

advantages and selecting the best after iteration is x_4 . Then, the time complexity of the algorithm at this stage is

$$O(N(nx_3 + f(n)) + x_4 + x_5 + x_6) = O(n + f(n)). \quad (26)$$

Therefore, the total time complexity of the ABC-KELM algorithm is

$$T(n) = O(n + f(n)) + O(n + f(n)) = O(n + f(n)). \quad (27)$$

In the IABC-KELM algorithm, the time required for the initialization phase of the algorithm is basically the same as the ABC-KELM algorithm. Therefore, the time complexity of the initialization phase of the improved algorithm is the same as equation (25). In the algorithm loop, suppose the calculation time of the weighted center is z_1 , the calculation time of the individual learning position is z_2 , and the calculation time of the comparison and selection process between the learning individual and the initial individual is z_3 . Then, the time complexity of the loop part is

$$O(N(nx_3 + f(n)) + x_4 + x_5 + x_6 + N(z_2 + z_3) + z_1) = O(n + f(n)). \quad (28)$$

Therefore, the total time complexity of the IABC-KELM algorithm to solve the optimal of each generation is

$$T(n) = O(n + f(n)) + O(n + f(n)) = O(n + f(n)). \quad (29)$$

In summary, the improved strategy of the IABC-KELM algorithm does not increase the time complexity of the algorithm solution compared to the initial ABC-KELM algorithm.

5. Comparison and Analysis of Algorithm Simulation

In order to obtain the output of the sensor component of the faulty node, it takes a lot of time and cost to do experiments due to the acquisition of a large amount of the fault data. Most of the existing literature uses Simulink simulation to obtain the fault output data. We refer to the use of Simulink in MATLAB to simulate the hardware fault of the sensor node and use it for fault diagnosis and prediction of the sensor node. The mathematical model is used to describe the output voltage signal changes after the occurrence of four types of common faults, including the impact faults, bias faults, short-circuit faults, and offset faults, which are prone to occur in the sensor nodes of WSNs.

Based on this, a fault diagnosis algorithm for sensor nodes based on kernel extreme learning machine optimized by artificial bee colony is proposed to diagnose hardware faults of the sensor nodes.

The steps of the node fault diagnosis method of the wireless sensor network based on IABC-KELM mainly include

- (1) Establish a fault diagnosis decision table for the wireless sensor network

FIGURE 4: The flowchart of the IABC-KELM.

- (2) Use discernibility matrix reduction algorithm based on core attributes to reduce redundant attributes
- (3) Extract fault diagnosis rules based on the simplest attributes
- (4) Establish the KELM fault diagnosis model according to the diagnosis rules and use the artificial bee colony algorithm to improve and optimize it, so that the improved algorithm has the best classification effect. Then, use the optimized kernel extreme learning machine algorithm to diagnose the hardware fault of the sensor node. The flowchart of the fault diagnosis strategy for the sensor nodes of WSNs based on IABC-KELM is shown in Figure 5

The important parameter regularization factor C and the kernel parameter s of the multicore function in the ABC-KELM algorithm are calculated by the artificial bee colony algorithm optimization iterative calculation. In order to prevent the ABC algorithm from overfitting and local optimal solution problems, and to obtain unbiased performance evaluation that we do not want, we introduce a disturbance mechanism into the algorithm to improve the ability of the ABC algorithm to jump out of the local optimal solution. In order to increase the diversity of the population, Cauchy mutation is performed on the n optimal individuals with a certain probability p , and then the optimal individuals after mutation are optimized twice, which endows the algorithm with the ability to jump out of the local optimal solution.

FIGURE 5: Fault diagnosis flowchart of the sensor node based on IABC-KELM.

TABLE 2: Pseudocode of the IABC-KELM algorithm.

Serial	Pseudocode of the IABC-KELM algorithm
1	The bee colony algorithm is initialized, the population size is SN, the maximum number of iterations is MCN, the learning period is L , the learning coefficient is λ , the probability is p_i , and the location information of all populations are initialized
2	The fitness value of each employed bee is calculated, and the global optimal fitness value is saved.
3	While ($k < MCN$)
4	For $i = 1$ to SN
5	For all the employed bees in the colony, according to the corresponding the probability p_i , a suitable strategy is selected by the roulette selection method
6	Update location information v_{ij} according to the selected strategy
7	Calculate the current fitness value $f(v_i)$
8	If ($f(v_i) > f(x_i)$)
9	Set $v_i = x_i$
10	Else if r and $< P$
11	Calculate the position information of the current colony
12	Calculate the current fitness value $f(x_i)$
13	If ($f(v_i) > f(x_i)$)
14	Set $v_i = x_i$
15	End if
16	End if
17	End for
18	[maxfitness index] = max ($f(x_i)$)
19	According to the fitness value $f(x)$ in descending order [$f(x)_1, f(x)_2, f(x)_3, \dots, f(x)_{NP}$]
20	For $i = 1$ to SN
21	Find the j th strategy to generate the location information v_j
22	If $k \% L = 0$
23	Calculate the probability value according to formula (17)
24	End if
25	Use formula (18) to calculate the probability value
26	End for
27	The next iteration until the termination condition is met
28	End while
29	Obtain the optimal regularization coefficient C and the kernel parameter s in the global optimal solution
30	The obtained parameter values are used for kernel extreme learning machine training
31	Test fault diagnosis sample data. Input: data sample set X and algorithm parameter initialization, output: failure accuracy result
32	Start
33	For $i = 1$ to SN
34	Use IABC-KELM to obtain the optimal parameters (C, s) on X_{train} to train the nuclear extreme learning machine model. Perform the node fault diagnosis test of HWSNs on X_{test}
35	End for
36	Returns the average fault diagnosis accuracy rate of the fault diagnosis model on the node fault diagnosis test data set X_{test}
37	End

The improved artificial bee colony algorithm is used to optimize the parameter regularization factor C and the kernel parameter s of the kernel extreme learning machine model, and the obtained optimal parameter set is used to train the kernel extreme learning machine model. Finally, use the test set to test on the trained fault diagnosis model to obtain the node fault diagnosis results of HWSNs.

The main steps of node fault diagnosis of HWSNs based on the IABC-KELM algorithm are as follows:

Step 1. The node fault diagnosis data set of HWSNs is divided into training data set and test data set. Seventy percent of the node fault diagnosis data set is used as the training set, and the remaining 30% is used as the test set.

Step 2. Randomly generate the initial population and encode it. The parameters of the initialization algorithm include the number of artificial bee colonies SN, the maximum number of iterations MCN, the learning period L , the learning coefficient λ , the probability p_i , and the position information of all the populations that are initialized.

Step 3. Using the solution vector obtained in step 2 for decoding, the regularization factor C and the kernel parameter s can be obtained, which are used to train the multi-core extreme learning machine model.

Step 4. Increase the number of iterations, $\text{iter} = \text{iter} + 1$.

Step 5. According to the execution probability p_i , update the optimal solution from the update strategy. In the first iteration, the probability of each solution update strategy being selected is equal, and the probability p_i is 0.25.

Step 6. Use the selected solution update strategy to update the solution.

Step 7. Use the parameter set obtained in step 6 to train the kernel extreme learning machine model on the training set and calculate the fitness value corresponding to each solution.

Step 8. Update the individual optimal fitness value and individual optimal position by comparing the current fitness value and the solution position. If the current fitness value is less than the fitness value in the historical record, the fitness value of the historical record is kept unchanged. Otherwise, generate a new solution according to the probability that changes with the number of iterations and go to step 7 to run; finally, go to step 9 to run.

Step 9. By comparing the current fitness value with the optimal fitness value in the historical record, the global optimal fitness value and the global optimal solution are updated. If the currently obtained optimal fitness value is less than the fitness value in the historical record, the historical record value will be kept unchanged; otherwise, it will replace the recorded fitness value.

Step 10. According to the result of the i th solution update strategy phase, formula (17) and formula (18) are used to update the probability value of each strategy.

Step 11. If the algorithm meets the termination condition (if the iteration reaches the set maximum value), go to step 12; otherwise, go to step 4 to continue the algorithm.

Step 12. Obtain the optimal regularization factor C and the kernel parameter s from the global optimal solution.

Step 13. The parameter values (C, s) obtained in step 12 are used for nuclear extreme learning machine training and tested on the fault diagnosis test data set, and the average

FIGURE 6: Function fitting iterative calculation results of the four algorithms.

value of the calculation results is taken to finally obtain the node fault diagnosis result of HWSNs.

The pseudocode based on the IABC-KELM algorithm is shown in Table 2.

6. Simulation Results and Analysis

6.1. Simulation Environment Settings. In order to verify the performance of the improved IABC-KELM algorithm in terms of convergence and optimization speed, the proposed IABC-KELM algorithm is compared with PSO-KELM algorithm, GWO-KELM algorithm, and ABC-KELM algorithm on function test and regression and classification data sets, respectively. The maximum number of population evolution in all experiments is set to 50, and the population size of the algorithm is 30. All experiments were run 50 times, and the root mean square error or the average and standard deviation of the classification accuracy were taken as the experimental results. The parameter settings of the PSO algorithm are as follows: learning factor $c_1 = 2$, $c_2 = 2$, inertia weight factor $\omega_1 = 0.9$, $\omega_2 = 0.4$, the number of particles in the initial population is 30, and the maximum number of iterations is $T_{\max} = 1000$. The GWO algorithm sets the gray wolf population size to 30, the control parameters are $\alpha_1 = 0.9$ and $\alpha_2 = 0.4$, and the maximum number of iterations of the algorithm $T_{\max}$ is 1000. The parameters of the ABC-KELM algorithm and the IABC-KELM algorithm are set as follows: the number of the bees population NP is 30, the number of hired bees SN = NP/2, and the maximum number of mining times is NP/2.

6.2. Test Objective Function Optimization. The simulation experiments in this section is carried out from the test objective function optimization (Sinc function fitting simulation

TABLE 3: Comparison of Sinc function fitting results.

Iteration	PSO-KELM			GWO-KELM			ABC-KELM			IABC-KELM		
	RMSE	MAE	RSD	RMSE	MAE	RSD	RMSE	MAE	RSD	RMSE	MAE	RSD
5	0.014	0.009	0.988	0.017	0.009	0.968	0.014	0.007	0.990	0.010	0.006	0.993
10	0.011	0.007	0.986	0.014	0.009	0.971	0.011	0.007	0.993	0.008	0.006	0.995
15	0.010	0.007	0.984	0.012	0.007	0.971	0.010	0.006	0.994	0.007	0.005	0.996
20	0.010	0.006	0.982	0.011	0.007	0.969	0.010	0.005	0.994	0.007	0.005	0.997
25	0.009	0.006	0.983	0.010	0.006	0.967	0.009	0.005	0.994	0.007	0.004	0.997
30	0.009	0.005	0.984	0.009	0.006	0.965	0.009	0.005	0.994	0.007	0.004	0.997
35	0.009	0.005	0.986	0.009	0.006	0.962	0.009	0.005	0.994	0.007	0.004	0.998
40	0.009	0.005	0.987	0.009	0.005	0.960	0.009	0.004	0.994	0.006	0.003	0.998
45	0.009	0.004	0.987	0.009	0.004	0.959	0.008	0.004	0.995	0.006	0.003	0.998
50	0.009	0.004	0.988	0.009	0.004	0.957	0.008	0.004	0.995	0.006	0.003	0.998

experiment, regression data set simulation experiment, and classification data set simulation experiment) and the sensor node hardware fault diagnosis of WSNs.

6.2.1. Sinc Function Simulation Experiment Comparison. The four algorithms are compared by fitting the Sinc function. The expression of the Sinc function is as follows:

$$f(x) = \begin{cases} \frac{\sin(x)}{x}, & x \neq 0 \\ 0, & x = 0 \end{cases}. \quad (30)$$

We set to generate 1000 data sets $\times$ uniformly distributed from $[-10,10]$ and calculate 1000 data sets $\{x_i, f(x_i)\}$, $i = 1, 2, 3, \dots, 1000$. Then, generate 1000 uniformly distributed noises ε ranging from $[-0.2,0.2]$. Let the training set be $\{x_i, f(x_i) + \varepsilon_i\}$, $i = 1, 2, 3, \dots, 1000$ and then generate another group of 1000 data sets $\{y_i, f(y_i)\}$, $i = 1, 2, 3, \dots, 1000$ as the test set. Gradually increase the number of iterations of the four algorithms to fit the function. The ABC-ELM and IABC-ELM algorithm parameter settings are the same. The function fitting iterative calculation results of the four algorithms are shown in Figure 6.

It can be seen from Figure 6 that as the number of simulation iterations increases, the root mean square error of the function fitting of the four algorithms is gradually shrinking. It can also be seen that the fitting error based on the PSO-KELM algorithm is the largest, the fitting error based on the GWO-KELM algorithm is larger, and the fitting error based on the ABC-KELM algorithm is smaller. The IABC-KELM algorithm proposed in this paper has the smallest fitting error and the best fitting effect.

In addition, root mean square error (RMSE), mean absolute error (MAE), and relative standard deviation (RSD) indicators are used as evaluation indicators for error analysis. The calculation formulas of the three indicators are as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y(i) - y'(i))^2},$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y'(i) - y(i)| = \frac{1}{N} \sum_{i=1}^N |e_i|,$$

$$\text{RSD} = \frac{\sqrt{\sum_{i=1}^N (y'(i) - \bar{y}'(i))^2}}{\sqrt{\sum_{i=1}^N (y(i) - \bar{y}(i))^2}}. \quad (31)$$

Among them, the parameter $y(i)$ represents the measured value, $y'(i)$ represents the predicted value, the parameter N is the number of samples, the parameter $e_i = y'(i) - y(i)$ is the absolute error, and the numerator and denominator of RSD are in the form of standard deviation. Among them, the smaller the RMSE and MAE index values, the lower the prediction error, and the closer the RSD index value is to 1, indicating that the predicted value is closer to the measured value. The Sinc function fitting results are shown in Table 3.

It can be seen from Table 2 that calculated by the PSO-KELM algorithm, the index values of RMSE and MAE are the largest, and the index value of RSD is the smallest, and the test result performance is poor. The RMSE and MAE index values of the GWO-KELM algorithm are larger, and the index value of RSD is even smaller, and the test result performance is average. The RMSE and MAE index values of the ABC-KELM algorithm are smaller, and the RSD index value is more connected, and the test result performance is better. The RMSE and MAE index values of the IABC-KELM algorithm are the smallest, the index value of RSD is the largest, and the test result performance is the best. It shows that the error of the IABC-KELM model is relatively smaller, and the prediction accuracy is better than that of the PSO-KELM, GWO-KELM, and ABC-KELM algorithms. At the same time, it can be known from the data change trend of Table 2 that the IABC-KELM algorithm has the best performance. Optimizing the KELM regularization parameter C and the kernel function s can improve the prediction accuracy of the KELM model.

6.2.2. Comparison of Regression Data Set Simulation Experiment. In this section, we use 4 real regression data sets in the machine learning library of the University of

TABLE 4: Comparison of the regression data set fitting results of the four algorithms.

Data Method	Auto MPG			Housing			Servo			Yacht hydrodynamics		
	RMSE	MAE	RSD	RMSE	MAE	RSD	RMSE	MAE	RSD	RMSE	MAE	RSD
PSO-KELM	0.0968	19.707	0.663	0.069	218.431	0.891	0.191	16.454	0.715	0.0393	1.797	0.898
GWO-KELM	0.0921	18.659	0.721	0.067	17.927	0.899	0.186	15.941	0.728	0.0352	1.705	0.921
ABC-KELM	0.0896	18.111	0.755	0.064	17.163	0.911	0.183	15.526	0.739	0.0332	1.736	0.938
IABC-KELM	0.0879	17.769	0.776	0.061	16.566	0.927	0.179	15.199	0.748	0.0318	2.011	0.961

TABLE 5: Comparison of experimental results of the classification data sets of four algorithms.

Data set	Blood transfusion		Ecoli		Iris		Wine	
	Accuracy	Std. Dev.	Accuracy	Std. Dev.	Accuracy	Std. Dev.	Accuracy	Std. Dev.
PSO-KELM	79.67%	0.008	90.47%	0.022	93.89%	0.015	91.13%	0.020
GWO-KELM	80.36%	0.009	90.57%	0.028	94.11%	0.027	91.85%	0.028
ABC-KELM	80.81%	0.013	90.78%	0.019	94.35%	0.019	92.15%	0.023
IABC-KELM	81.26%	0.014	91.67%	0.027	95.56%	0.016	93.54%	0.025

(a) Short-circuit fault

(b) Impact fault

(c) Bias fault

(d) Offset fault

FIGURE 7: Four kinds of hardware fault simulation.

California, Irvine, to compare the performance of the 4 algorithms. The names of the data sets are as follows: auto MPG (MPG), computer Hardware (CPU), housing, and servo. In the experiment, the data in the data set is randomly divided into training set and test set, 70% of which is used as training set and the remaining 30% is used as test set. In order to reduce the influence of the large difference of each variable,

we normalize the data before the algorithm runs; that is, the input variable is normalized to $[-1,1]$, and the output variable is normalized to $[0,1]$. In all experiments, the number of algorithm iterations is 50, and the average value of 50 experimental results is calculated. The experimental results of the regression data set of the four algorithms are shown in Table 4.

FIGURE 8: Comparison of node hardware fault diagnosis results of four algorithms.

From the grid in Table 3, we can see that IABC-ELM has obtained the smallest RMSE, smallest MAE, and largest RSD value in the four data set fitting experiments. The performance of the algorithm is the best, and the performance of the algorithm is also the most stable. Judging from the results obtained by the number of iterations of 50 times, the convergence speed of PSO-KELM, GWO-KELM, and ABC-KELM is slow, and the calculated results are also poor. The accuracy of the fitting results calculated by the IABC-ELM algorithm proposed in this paper is the best. Therefore, among different data sets, the algorithm proposed in this paper can reach the smallest RMSE. Taking into account comprehensively, the performance of HABC-ELM is superior.

6.2.3. Comparison of Classification Data Set Simulation Experiments. In this experiment, we again used the machine learning library of the University of California, Irvine. The

names of the four real classification data sets are as follows: Blood Transfusion Service Center (Blood), E. coli, iris, and wine. Same as the classification data set, 70% of the experimental data is used as the training set and 30% as the test set. The input variables of the data set are normalized to $[-1,1]$. In the experiment, the number of iterations of the algorithm gradually increased, and the accuracy and standard deviation (Std. Dev.: standard deviation) experimental results of the four algorithms are shown in Table 5.

From the results in Table 5, the classification accuracy of Blood Transfusion Service Center (Blood) of the four algorithms is above 80%, and the classification accuracy of E. coli, iris, and wine is above 90%. On the whole, among the four algorithms, the classification accuracy of the PSO-KELM algorithm is low, the classification accuracy of the GWO-KELM algorithm is medium, and the classification accuracy of the ABC-KELM algorithm is high. The IABC-KELM

FIGURE 9: Comparison results of test set fault diagnosis.

algorithm proposed in this paper has the best classification accuracy, which is 2.3%, 1.7%, and 1.3% higher than the average accuracy of the other three algorithms.

From the simulation results of function testing, regression, and classification data sets, the IABC-KELM algorithm proposed in this paper has the highest accuracy, the ABC-KELM algorithm has higher accuracy, and the GWO-KELM algorithm has poor classification accuracy. The accuracy of the PSO-KELM algorithm is the worst. It can be seen that the IABC-KELM algorithm proposed in this paper has the best performance in terms of function testing, regression fitting and data classification, and the fastest finding the optimal solution.

6.3. Comparative Analysis of Fault Diagnosis. At present, there are four main types of faults in WSNs: impact fault, offset fault, short-circuit fault, and offset fault. This paper mainly studies these four types of common faults. In order to obtain the output of the sensor component of the faulty node, due to the acquisition of a large amount of fault data, it takes time and cost to do experiments. Most of the existing literature uses Simulink simulation to obtain the fault output data. We refer to Simulink in MATLAB to simulate the hardware failure of the sensor node and use it for later failure prediction. A mathematical model is used to describe the output voltage signal changes after the occurrence of four common faults: impact fault, bias fault, short-circuit fault, and offset fault. Assuming that the true value of the output voltage of a sensor node is $\gamma(t)$, and $\varepsilon(t)$ is the measurement noise of the sensor node, the node has a measurement error, and the measurement error conforms to a normal distribution. When the sensor node fails, the output voltage value can be expressed as $f(t) = \beta_0(t) + \beta_1(t) \times \gamma(t) + \varepsilon(t)$, the parameter $\beta_0(t)$ represents the offset value, and the parameter $\beta_1(t)$ represents the zoom factor. According to this mathematical model, four kinds of fault expressions are set, and Simulink is used to simulate the output signal of the sensor node.

FIGURE 10: Comparison of fault diagnosis accuracy rates of four algorithms.

FIGURE 11: Comparison of time-consuming simulation of four algorithms with different population sizes.

Four kinds of hardware fault simulation are shown in Figure 7. The simulation time is set to 200 s, and the occurrence time of the four types of faults is set to 120 s. In order to simulate the actual signal more realistically, random noise is added to the signal collected by the sensor. The failure of the sensor node will cause the signal to change suddenly, and the sudden change of the signal shows a change in the mean value. Through simulation, the output signal of the sensing node under four types of faults is as follows: impact fault, bias fault, short circuit fault, and offset fault. Run the simulation module 50 times and sample the output results of each time to obtain samples of four kinds of node hardware faults: short-circuit fault, impact fault, bias fault, and offset fault. Figure 6 shows the simulation signal output of

four types of faults: short-circuit fault, impact fault, bias fault, and offset fault. There are 50 groups of each type of the failure data sample, which has a total of 200 groups. Each sample is composed of 10 sampling values, of which 40 groups of each sample are used for training and 10 groups are used for fault prediction.

The steps of establishing the fault diagnosis model of the node hardware is as follows: first, preprocess the collected 200 sets of node fault data and divide it into a training sample set and a test sample set. Then, use the IABC-KELM algorithm to train the training sample set and get the fault prediction model based on IABC-KELM after training. Then, the trained model is used to predict the data of the test sample set, and the hardware failure of the sensor node is predicted and classified. The hardware fault diagnosis results of the wireless sensor network nodes of the four algorithms are shown in Figure 8.

Figure 8 is a comparison of the classification results of the test set. Figure 8 contains 60 test set samples represented by circles and the prediction results of the four classification models. The PSO-KELM algorithm and the GWO-KELM algorithm have good generalization performance and are different from the KELM neural network, their weights and thresholds are determined by training samples in one step, the calculation is iterative, and the amount of calculation is small. It can be seen that the four algorithms have obtained good prediction results. The final prediction result of PSO-KELM is 93.3%, the final prediction result of GWO-KELM algorithm is 95%, and the final prediction result of ABC-KELM is 96.7%. The final result of the IABC-KELM prediction algorithm proposed in this paper is 98.3%.

In order to give a more vivid comparison of the fault diagnosis results of the four algorithms, the specific fault diagnosis results of 60 test set samples are given. The specific fault diagnosis comparison results are shown in Figure 9. Figure 10 shows the comparison of the accuracy of the wireless sensor network fault diagnosis of the four algorithms.

It can be seen from Figure 9 that in the 60 test set fault data sets, the fault diagnosis method based on PSO-KELM is wrong 4, and the fault diagnosis method based on GWO-KELM is wrong 3. Two fault diagnosis methods based on ABC-KELM are wrong, and one fault diagnosis method based on IABC-KELM is wrong. The algorithm proposed in this paper has the lowest error rate in fault diagnosis. It can be seen from Figure 10 that the fault diagnosis results of the four algorithms are all above 90%. Among them, the fault diagnosis method based on PSO-KELM has the lowest average accuracy, the fault diagnosis method based on GWO-KELM is lower, and the fault diagnosis method based on ABC-KELM is higher. The fault diagnosis method based on IABC-KELM proposed in this paper is the highest.

In order to more comprehensively reflect the performance of the four algorithms, the simulation time comparison of the four algorithms is given, as shown in Figure 11.

It can be seen from Figure 11 that no matter the initial population size of the intelligent algorithm changes from 30, 40, or 50, the simulation time of the four algorithms is about 2 s, with little difference, and they are all within the acceptable fault diagnosis time range.

Based on the above experiments, compared with the other three fault prediction classification methods, the IABC-KELM fault diagnosis method proposed in this paper has the highest diagnosis accuracy, the best performance of the algorithm, and the best performance. The IABC-KELM proposed in this paper can compare and is well suitable for the hardware fault diagnosis of heterogeneous wireless sensor network nodes.

7. Conclusion

The problem of node fault diagnosis in WSNs is an urgent problem in the key technology of HWSNs. This paper firstly analyzes the fault classification and fault causes of HWSNs nodes. The regularization coefficient C and the kernel parameter s in KELM are used as model parameters to affect the accuracy of the fault diagnosis model of the nuclear extreme learning machine. A fault diagnosis method for HWSN nodes based on an improved artificial bee colony algorithm and improved nuclear extreme learning machine (ABC-KELM) is proposed. Finally, the effect is verified by experiments. The experiment shows that the improved fault diagnosis method of nuclear extreme learning machine has better fault accuracy than the extreme learning machine and neural network. The proposed algorithm improves the accuracy of sensor node hardware fault diagnosis and can be better applied to HWSNs node hardware fault diagnosis.

In the next work, we will study the failures of other nodes in the wireless sensor network, while using new swarm intelligence optimization algorithms and machine learning methods.

Data Availability

The data used to support the findings of this study are included in the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported in part by the Natural Science Foundation of Hubei Province under Grant 2020CFB304, the Open Found of Distinctive Disciplines in Hubei University of Arts and Science under Grant XK2020047, and the Educational Science Planning Issue of Hubei Province under Grant 2020GA057, in part by the talent introduction project of Oujiang College of Wenzhou University.

References

- [1] L. Cao, Y. Cai, and Y. Yue, "Data fusion algorithm for heterogeneous wireless sensor networks based on extreme learning machine optimized by particle swarm optimization," *Journal of Sensors*, vol. 2020, Article ID 2549324, 17 pages, 2020.
- [2] Y. Yue and P. He, "A comprehensive survey on the reliability of mobile wireless sensor networks: taxonomy, challenges, and future directions," *Information Fusion*, vol. 44, no. 11, pp. 188–204, 2018.

- [3] W. Deng, S. Shang, X. Cai et al., "Quantum differential evolution with cooperative coevolution framework and hybrid mutation strategy for large scale optimization," *Knowledge-Based Systems*, vol. 224, p. 107080, 2021.
- [4] X. Cai, H. Zhao, S. Shang et al., "An improved quantum-inspired cooperative co-evolution algorithm with multi-strategy and its application," *Expert Systems with Applications*, vol. 171, p. 114629, 2021.
- [5] W. Deng, J. Xu, X. Z. Gao, and H. Zhao, "An enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, no. 11, pp. 1–10, 2020.
- [6] W. Deng, J. Xu, H. Zhao, and Y. Song, "A Novel Gate Resource Allocation Method Using Improved PSO-Based QEA," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, pp. 1–9, 2020.
- [7] S. Guo, X. Zhang, Y. du, Y. Zheng, and Z. Cao, "Path planning of coastal ships based on optimized DQN reward function," *Journal of Marine Science and Engineering*, vol. 9, no. 2, pp. 210–219, 2021.
- [8] J. Wang, Y. Cao, B. Li, H.-J. Kim, and S. Lee, "Particle swarm optimization based clustering algorithm with mobile sink for WSNs," *Future generations computer systems*, vol. 76, pp. 452–457, 2017.
- [9] J. Wang, C. Ju, Y. Gao, A. K. Sangaiah, and G. J. Kim, "A PSO based energy efficient coverage control algorithm for wireless sensor networks," *Computers, Materials and Continua*, vol. 56, no. 3, pp. 433–446, 2018.
- [10] N. R. Sivakumar, "Stabilizing energy consumption in unequal clusters of wireless sensor networks," *Computers, Materials & Continua*, vol. 64, no. 1, pp. 81–96, 2020.
- [11] 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 and Continua*, vol. 58, no. 2, pp. 649–667, 2019.
- [12] K. Vijayalakshmi and P. Anandan, "Global levy flight of cuckoo search with particle swarm optimization for effective cluster head selection in wireless sensor network," *Intelligent Automation and Soft Computing*, vol. 26, no. 2, pp. 303–311, 2020.
- [13] N. Shahriar, R. Ahmed, S. R. Chowdhury, A. Khan, R. Boutaba, and J. Mitra, "Generalized recovery from node failure in virtual network embedding," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 261–274, 2017.
- [14] J. Li, "Data transmission scheme considering node failure for blockchain," *Wireless Personal Communications*, vol. 103, no. 1, pp. 179–194, 2018.
- [15] R. Ghimire, C. Zhang, and K. R. Pattipati, "A rough set-theory-based fault-diagnosis method for an electric power-steering system," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 5, pp. 2042–2053, 2018.
- [16] M. M. M. Islam and J. M. Kim, "Automated bearing fault diagnosis scheme using 2D representation of wavelet packet transform and deep convolutional neural network," *Computers in Industry*, vol. 106, no. 5, pp. 142–153, 2019.
- [17] P. Gangsar and R. Tiwari, "A support vector machine based fault diagnostics of Induction motors for practical situation of multi-sensor limited data case," *Measurement*, vol. 135, no. 4, pp. 694–711, 2019.
- [18] Y. Cheng, Q. Liu, J. Wang, S. Wan, and T. Umer, "Distributed fault detection for wireless sensor networks based on support vector regression," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 4349795, 8 pages, 2018.
- [19] S. K. Preeth, R. Dhanalakshmi, and R. Kumar, "An adaptive fuzzy rule based energy efficient clustering and immune-inspired routing protocol for WSN-assisted IoT system," *Journal of Ambient Intelligence and Humanized Computing*, vol. 44, no. 2, pp. 1–13, 2018.
- [20] Y. Zhang, Y. Wang, G. Zhou et al., "Multi-kernel extreme learning machine for EEG classification in brain-computer interfaces," *Expert Systems with Applications*, vol. 96, no. 4, pp. 302–310, 2018.
- [21] H. Chen, Q. Zhang, J. Luo, Y. Xu, and X. Zhang, "An enhanced bacterial foraging optimization and its application for training kernel extreme learning machine," *Applied Soft Computing*, vol. 86, no. 1, pp. 105884–105897, 2020.
- [22] M. Wang, H. Chen, H. Li et al., "Grey wolf optimization evolving kernel extreme learning machine: application to bankruptcy prediction," *Engineering Applications of Artificial Intelligence*, vol. 63, no. 8, pp. 54–68, 2017.
- [23] W. Zhu, C. Ma, X. Zhao et al., "Evaluation of Sino foreign cooperative education project using orthogonal sine cosine optimized kernel extreme learning machine," *IEEE Access*, vol. 8, no. 3, pp. 61107–61123, 2020.
- [24] Y. Zhang, S. Cheng, Y. Shi, D. W. Gong, and X. Zhao, "Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm," *Expert Systems with Applications*, vol. 137, no. 12, pp. 46–58, 2019.
- [25] J. Li, M. Song, L. Wang et al., "Hybrid artificial bee colony algorithm for a parallel batching distributed flow-shop problem with deteriorating jobs," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2425–2439, 2020.
- [26] J. Zhou, X. Yao, F. Chan et al., "An individual dependent multi-colony artificial bee colony algorithm," *Information Sciences*, vol. 485, no. 6, pp. 114–140, 2019.
- [27] A. A. Hady, "Duty cycling centralized hierarchical routing protocol with content analysis duty cycling mechanism for wireless sensor networks," *Computer Systems Science and Engineering*, vol. 35, no. 5, pp. 347–355, 2020.
- [28] ShayestehTabatabaei, "A novel fault tolerance energy-aware clustering method via social spider optimization (SSO) and fuzzy logic and Mobile sink in wireless sensor networks (WSNs)," *Computer Systems Science and Engineering*, vol. 35, no. 6, pp. 477–494, 2020.