

## Research Article

# Data Fusion Algorithm for Heterogeneous Wireless Sensor Networks Based on Extreme Learning Machine Optimized by Particle Swarm Optimization

Li Cao,<sup>1</sup> Yong Cai ,<sup>1</sup> and Yinggao Yue <sup>2</sup>

<sup>1</sup>School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China

<sup>2</sup>Computer School, Hubei University of Arts and Science, Xiangyang 441053, China

Correspondence should be addressed to Yinggao Yue; [yueyinggao2006@163.com](mailto:yueyinggao2006@163.com)

Received 19 September 2019; Revised 1 June 2020; Accepted 16 June 2020; Published 13 August 2020

Academic Editor: Mohammad Haider

Copyright © 2020 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.

Data fusion can reduce the data communication time between sensor nodes, reduce energy consumption, and prolong the lifetime of the network, making it an important research focus in the field of heterogeneous wireless sensor networks (HWSNs). Normal sensor nodes are susceptible to external environmental interferences, which affect the measurement results. In addition, raw data contain redundant information. The transmission of redundant information consumes excess energy, thereby reducing the lifetime of the network. We propose a data fusion method based on an extreme learning machine optimized by particle swarm optimization for HWSNs. The spatiotemporal correlation between the data of the HWSNs is determined, and the extreme learning machine method is used to process the data collected by the sensor nodes in the hierarchical routing structure of the HWSN. The particle swarm optimization algorithm is used to optimize the input weight matrix and the hidden layer bias of the extreme learning machine. An output weight matrix is created to reduce the number of hidden layer nodes and improve the generalization ability of the model. The data fusion model fuses the original data collected by the sensor nodes. The simulation results show that the proposed algorithm reduces network energy consumption and improves the lifetime of the network, the efficiency of data fusion, and the reliability of data transmission compared with other data fusion methods.

## 1. Introduction

The field of wireless sensor networks is an emerging discipline and a core technology of the perception layer of the Internet of Things [1]. Therefore, wireless sensor networks are widely used in intelligent transportation, environmental monitoring, intelligent medical objects, and other fields. Sensor nodes are needed to collect data in many cases, but the environment of the sensors may be harsh, and batteries can often not be replaced [2]. The use of heterogeneous nodes has become a core issue in wireless sensor network research because these nodes prevent network paralysis caused by the malfunction of sensor nodes. A large number of sensor nodes are commonly deployed for comprehensive information collection in the monitoring area [3]. The overlap in the area that the sensor nodes cover results in high data

redundancy obtained from the sensor nodes [4]. The volume of redundant information is massive, causing a waste of human and material resources [5, 6]. Heterogeneous wireless sensor networks (HWSNs) are commonly used for real-time monitoring of environmental data, which contain random errors [7, 8]. The data collected by each sensor are processed to obtain new and more close-to-target information, i.e., data fusion is performed [9]. The advantages of data fusion include high accuracy of information acquisition, energy conservation, and high efficiency of data collection. The ultimate goal is to take advantage of a multisensor operation to improve the effectiveness of the system [10–12]. The objective of data fusion is to minimize or eliminate redundant, convergent, and other operations so that the data collected from multiple sensor nodes is concise and accurate [13]. Moreover, data fusion reduces the number of packet

transmissions and the power consumption of the network, thereby improving the data collection efficiency and creating a highly fault-tolerant system [14].

To avoid potential problems, different types of data sent by the cluster nodes to the cluster head nodes in HWSNs must be merged, followed by comprehensive analysis and processing. Invalid, redundant, and poor data are deleted to reduce the volume of redundant data for transmission and improve the accuracy of the data. Data fusion research is of significance because data fusion has the following advantages:

- (1) Reduce energy consumption. Data fusion and redundant information processing are achieved as follows: before the intermediate node sends data, it performs processing steps on the data of the sensor nodes, such as removing redundant data, compressing data, processing data hierarchically, and evaluating the operation. Without losing content, the transmitted data volume is minimized, which reduces energy consumption
- (2) Enhance data security. It is difficult to ensure the accuracy of the information for a small volume of node data. Generally, the target data are collected by multiple sensor nodes to improve the accuracy and reliability of the information. If the data collected by an individual node differs significantly from that of the other nodes, the data can be excluded using a simple comparison algorithm
- (3) Reduce network delay and improve transmission efficiency. Data fusion in the network reduces the size of data packets to be transmitted, optimizes the transmission routes, and classifies the data; in addition, it reduces data collision conflicts, network congestion, and data transmission delays and improves bandwidth utilization, thereby improving the overall efficiency of the network
- (4) Optimize the network resources and improve the overall network performance. Since data fusion reduces energy consumption, it balances the energy consumption of the sensor nodes, avoids energy holes, and maximizes the lifetime of nodes, thereby improving the overall performance of the network

The data fusion algorithm improves the performance of HWSNs and the reliability of data transmission, but some performance degradation will occur. On the one hand, the transmission delay of the network is increased because the cluster head node requires additional computing time for the data fusion, and the data collected by the cluster member nodes are sent to the cluster head node, which results in additional data transmission time. On the other hand, the data fusion algorithm reduces the robustness of the network. Due to the high packet loss rate in the wireless communication of HWSNs, the information volume of the cluster nodes is greatly increased, and the data fusion causes further information loss, which reduces the robustness of the network. In this study, we make use of the spatiotemporal correlation in the data acquisition

by the sensor nodes and use data fusion to reduce the data transmission volume, improve the lifetime and efficiency of the network, and reduce the energy consumption of HWSNs.

An artificial intelligence-based data fusion approach for HWSNs is proposed. The spatiotemporal correlation between the data of the sensor nodes is considered, and an extreme learning machine (ELM) method is applied to process the data collected by the sensor nodes in the hierarchical routing structure of the HWSN. The proposed algorithm reduces the data transmission volume, as well as the processing time of the data. In summary, we make the following contributions:

- (1) We conduct an in-depth analysis of data fusion in HWSNs
- (2) We propose a new data fusion algorithm based on the ELM optimized by particle swarm optimization (PSO)
- (3) We conduct extensive simulations to demonstrate the validity and efficiency of the proposed data fusion algorithm based on the PSO-ELM method
- (4) We evaluate the performance of the proposed data fusion approach experimentally and compare it with three other approaches, i.e., the stable election protocol (SEP), the data fusion method based on BP neural network (BPDF), and the data fusion method based on extreme learning machine (ELMDF).

The rest of this paper is organized as follows: Section 2 presents an in-depth analysis of the existing approaches. Section 3 presents the proposed approach PSO-ELM method for HWSNs. In Section 4, we describe the system model and problem overview, as well as the proposed data fusion approach for HWSNs. In Section 5, the performance of the proposed approach is analyzed. Finally, the conclusions are presented in Section 6.

## 2. Related Work

There are four general categories of data fusion algorithms for HWSNs: data fusion combined with temporal correlation, data fusion combined with spatial correlation, data fusion combined with spatiotemporal correlation, and data fusion combined with a routing protocol. Experts and scholars have conducted numerous studies on data fusion in HWSNs and have made considerable progress.

*2.1. Data Fusion Combined with Temporal Correlation.* Since physical phenomena in the monitoring environment change over time, the measurements obtained by the sensor node are highly correlated. In an HWSN, the volume of redundant data is high if the sensor node frequently sends monitoring data. Temporal correlation is common for a single sensor node. The changes in a time series of the sensor data are analyzed with a mathematical model to adjust the sampling frequency adaptively or predict changes in the observation index. In [15], the authors proposed a novel data collection approach called the reliability and multipath encounter

routing method (RMER), which met the reliability and energy efficiency requirements of the network. The RMER approach sends the data to the sink by converging multipath routes of event monitoring nodes into a one-path route to aggregate the data. In [16], the authors proposed a novel framework that included data prediction, compression, and recovery to achieve accurate and efficient data processing in clustered WSNs. Xiao et al. proposed a collaborative sensor selection method named CSdT based on the sensor-target distance and the sensor correlation; the method provided the required energy balance and met the accuracy requirements [17].

**2.2. Data Fusion Combined with Spatial Correlation.** This method is used to minimize data redundancy in sensing nodes that have similar distances in the monitoring area. The objective of the algorithm is to assess the similarity of the sensing data of the nodes in the same monitoring area at the same time, to sense the data of the sensor nodes, or to represent the sensing data as a number indicating the state at a particular time and then upload the information to reduce the energy consumption of the network. In [18], the authors established a spatial correlation aware dynamic and scalable routing structure for data collection and aggregation in WSNs; the proposed algorithm provided better aggregation performance than other comparable algorithms. The authors determined critical spatial correlation characteristics to reduce the computational cost. The rate-distortion theory (RD) is a data aggregation technique that takes advantage of the spatial correlation using a cluster-based communication model; several low-overhead protocols were proposed based on this model [19]. In Ref. [20], the authors proposed the efficient data collection aware of spatiotemporal correlation (EAST), which uses the shortest routes and exploits spatial and temporal correlations to perform near real-time data collection in WSNs.

**2.3. Data Fusion Combined with Spatiotemporal Correlation.** This type of algorithm is primarily mainly applied to the monitoring of observation indicators that change regularly over time in the same monitoring area of HWSNs. The objective of the algorithm is to reduce the data redundancy by analyzing and processing the temporal and spatial changes in the data. Data fusion techniques have attracted the attention of many scholars, and many effective data fusion algorithms were proposed. In [21], a dynamics-decoupled, multisource capable energy model was presented, which could handle the fast random patterns of communications and energy harvesting. By reducing the redundancy of the spatiotemporal data, the authors exploited both the spatial and temporal correlation of the sensor data; the spatial redundancy of the sensor data was reduced by determining the similarity of the subclusters, and the temporal redundancy was reduced by a model-based prediction approach [22]. In [23], the authors proposed a spatiotemporal correlation method for data collection in WSNs based on the low-rank matrix approximation (LRMA). The temporal consistency and the spatial correlation of the data were simultaneously integrated into the LRMA model.

**2.4. Data Fusion Combined with Routing Protocol.** In the communication protocol layers, such as the physical layer, the data link layer, and the network layer of the HWSNs, data fusion methods consider different factors, and the implementation algorithms are also different. The data fusion algorithm is combined with the routing protocol and, according to different data sources, types, and contents; the intermediate node optimizes the data using the appropriate data fusion algorithm. Therefore, most of the information can be represented by a small amount of data, which reduces the data transmission volume. A multisensor data fusion approach for WSNs based on Bayesian methods and ant colony optimization techniques was proposed in [24]; the proposed algorithms reduced the energy use. In [25], the authors proposed the enhanced-optimized energy-efficient routing protocol (E-OEERP), which eliminated individual node formation and improved the network's lifetime compared to existing protocols. In Ref. [26], the authors presented a new multiagent system (MAS) specially designed to manage data in WSNs; the method was tested in a residential home for the elderly, and it was found that the information fusion processes were improved.

The combination of data fusion and routing protocols has advantages and disadvantages related to energy efficiency, scalability, communication bandwidth utilization, latency, and throughput; none of them can meet all the requirements of data fusion. In addition, as the size of HWSNs has increased, so has the complexity and latency of data fusion algorithms. It is difficult to meet the application requirements because of the limited resources of HWSNs. In this study, we review the energy consumption and network performance of different data fusion algorithms and propose a new data fusion method for HWSNs based on an ELM. In this algorithm, the weights and thresholds of the neural network are optimized using the ELM. The raw data collected by the sensor nodes are extracted and merged, and the cluster head node sends the fused data to the sink. The data fusion process eliminates the transmission of redundant data and improves the accuracy of the results. The proposed data fusion method greatly reduces the dimensionality of the data, prolongs the lifetime of the network, and improves the efficiency of data fusion and the reliability of data transmission.

### **3. Extreme Learning Machine Optimized by PSO Algorithm**

In this section, we describe the ELM algorithm and its optimization by the PSO method.

**3.1. Extreme Learning Machine.** An ELM is a fast and efficient learning algorithm that was first proposed by Professor G. Huang from Nanyang Technological University in Singapore in 2006 [27]. It is developed from a single hidden layer feed-forward neural network with fast learning speed and high generalization performance [28, 29]. The ELM network model is shown in Figure 1. The algorithm has been widely used and has received favorable reviews by many researchers, engineers, and technicians. Compared with other neural

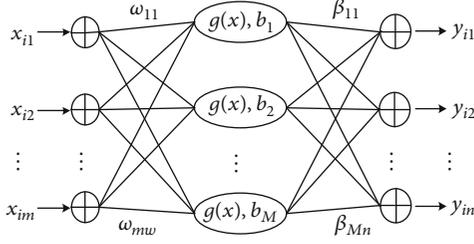


FIGURE 1: The model of extreme learning machine.

network algorithms, the ELM has the advantages of simplicity, easy implementation, high efficiency, and fast speed. The reason is the use of the kernel function in the ELM, which significantly reduces the complexity of the algorithm and provides excellent generalization performance and high stability, as well as an optimal solution [30]. The ELM is a regression and classification algorithm and has been widely used and improved in various fields.

The input layer receives the initial information and the input variables from the environment and passes this information on to the next layer, which is the hidden layer. The hidden layer only receives signals from the input layer and does not receive or transmit information to other layers. The purpose of the hidden layer is to process and identify the variables. Subsequently, the information is passed on to the output layer [31].

We assume that the parameter  $(m, n)$  represents the input weight and the threshold of each node of the hidden layer, and the training sample set is represented by  $(x, a)$ .  $f(x)$  is the expression of the mapping function of the hidden layer, the parameter  $\alpha$  is the output weight, the parameter  $M$  is the number of nodes of the hidden layer, and the parameter  $W_j$  is the error of the entire network, which is expressed as follows:

$$W_j = \left\| \sum_{i=1}^M \alpha_i f(m_i, n_i, x) - a \right\|. \quad (1)$$

The variable  $L$  represents the matrix at the time of the hidden layer output, and the following expression is expressed according to Eq. (1):

$$W = \|L \times \alpha - a\|. \quad (2)$$

The ELM continuously reduces the learning error in the neural network through continuous learning. When the error is 0, it indicates the optimal learning conditions of the ELM:

$$L \times \alpha = A. \quad (3)$$

If the variables  $L$  and  $A$  are known, the value of  $\alpha$ , which is the output weight, can be obtained from Eq. (3). This process represents the learning process of the ELM.

Assuming that the output matrix  $L$  is a nonsingular matrix, the parameter  $\alpha$  can be obtained from Eq. (3), and the training sample set of the ELM will error-free.

$$\alpha = L^{-1} \times a. \quad (4)$$

The parameter  $L^{-1}$  in Eq. (4) is the inverse matrix of the parameter  $L$ .

The model of the ELM can be expressed mathematically based on a random selection of the parameter  $R$  and the different samples  $(a_i, b_j)$ , where

$$a_i = [a_{i1}, a_{i2}, a_{i3}, \dots, a_{in}]^T, a_i \in R^n, \quad (5)$$

$$b_i = [b_{i1}, b_{i2}, b_{i3}, \dots, b_{im}]^T, b_i \in R^m. \quad (6)$$

An infinitely differentiable activation function  $f(x)$  in any interval is defined for a feed-forward neural network with the hidden layer neurons the parameter  $Z$ . The output model of the network is expressed as:

$$\sum_{i=1}^Z \varphi_i f_i(a_j) = \sum_{i=1}^Z \varphi_i f_i(\delta_i \times a_j + v_i) = o_j, \quad (7)$$

where the parameter  $j = 1, 2, 3, \dots, R$ , the parameter  $\delta_i = [\delta_{i1}, \delta_{i2}, \dots, \delta_{in}]^T$ , and the weight of the input node and the hidden layer node connecting the  $i$ -th sample are defined as:

$$\varphi_i = [\varphi_{i1}, \varphi_{i2}, \dots, \varphi_{im}]^T. \quad (8)$$

The parameter  $\varphi_i$  is the weight that connects the output node to the hidden layer node.  $o_j$  represents the output value of the  $j$ -th input sample, and the parameter  $v_i$  represents the threshold of the  $i$ -th hidden layer node.

If the number of the samples in the training set is equal to the number of hidden layer neurons, then for a random  $\delta$  and  $v$ , the feed-forward neural network can approach the training sample with a difference of the zero error.

$$\sum_{j=1}^Z \|o_j - y_j\| = 0. \quad (9)$$

Equation (10) is obtained:

$$\sum_{j=1}^Z \varphi_i f_i(\delta_i \times x_j + v_i) = y_j, \quad (10)$$

where the parameter  $j = 1, 2, 3, \dots, R$ ; according to  $F \times \varphi = Y$ , Eq. (11) is obtained:

$$Y = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_n^T \end{bmatrix}_{R \times m}, \varphi = \begin{bmatrix} \varphi_1^T \\ \varphi_2^T \\ \vdots \\ \varphi_n^T \end{bmatrix}_{Z \times m}. \quad (11)$$

The variable  $F$  is the hidden layer output matrix, which is expressed as:

$$F(\delta_1, \delta_2, \dots, \delta_M, v_1, v_2, \dots, v_M, x_1, x_2, \dots, x_M) = \begin{bmatrix} f(\delta_1 \times x_1 + v_1) & f(\delta_2 \times x_1 + v_2) & \dots & f(\delta_M \times x_1 + v_Z) \\ f(\delta_1 \times x_2 + v_1) & f(\delta_2 \times x_2 + v_2) & \dots & f(\delta_M \times x_2 + v_Z) \\ \vdots & \vdots & \dots & \vdots \\ f(\delta_1 \times x_R + v_1) & f(\delta_2 \times x_R + v_2) & \dots & f(\delta_M \times x_R + v_Z) \end{bmatrix} \quad (12)$$

Typically, the number of neurons in the hidden layer  $Z$  is lower than the parameter  $R$  so that the computational cost is low for a large number of training samples. The training error of the feed-forward neural network is an arbitrary value at this time;  $\varepsilon > 0$ ,  $\sum_{j=1}^Z \|o_j - y_j\| < \varepsilon$ . When the activation function  $f(x)$  is divisible, the parameters in the feed-forward neural network do not need to be adjusted and remain unchanged throughout the training process.

However, due to the random selection of the input weights and biased values of the hidden layers, several non-optimal parameters will be generated during the learning process, which results in high norm values of the output weights and condition numbers of the output matrix of the hidden layers, adversely affecting the generalization ability of the hidden layers. The traditional optimization method only focuses on minimizing the root mean square error (RMSE) but does not consider the output weight norm and the hidden layer output of the matrix condition number, which causes network instability. The PSO algorithm is used to optimize the input weight matrix and hidden layer offset of the ELM, and the output weight matrix is calculated to reduce the number of hidden layer nodes and improve the generalization ability of the model.

**3.2. Particle Swarm Optimization Algorithm.** The PSO algorithm is a population-based stochastic optimization technique proposed by Kennedy and Eberhart in 1995 [32]. The original algorithm was used to simulate the foraging behavior of birds. As an evolutionary algorithm, the PSO algorithm is straightforward with simple implementation and few parameters that require adjustment. It is an effective global optimal search algorithm that is widely used in engineering and social sciences [33].

In the PSO, the initial feasible solution for each execution problem is randomly set, and each particle is a candidate solution in the  $d$ -dimensional flight space [34]. All the particles in the particle swarm have random initial velocity and position vectors. The velocity vector allows the particle to search for a feasible solution in the search space and find the global optimal position vector after several iterations. The position vector of the  $i$ -th particle at the  $t$ -th iteration is represented as  $X_i(t) = [x_{i1}, x_{i2}, \dots, x_{id}]$ , and the speed vector table is defined as  $V_i(t) = [v_{i1}, v_{i2}, \dots, v_{id}]$ . The fitness function (which is maximized or minimized) is the optimization goal of the PSO and is defined as the proximity of a particle to the optimal position. The PSO algorithm calculates the fitness value of each iteration and finds the two optimal vectors  $P_{ib} = [P_{i,1}, P_{i,2}, \dots, P_{i,d}]$  and  $P_G = [P_{G,1}, P_{G,2}, \dots, P_{G,d}]$  during the iteration. The former is the optimal position vector

through which the  $i$ -th particle passes, and the latter is the optimal vector through which the entire population passes. In each iteration, the particles adjust the velocity vector according to the effects of the parameter  $P_{ib}$  and the parameter  $P_G$  so that the particles of the PSO can search for the optimal solution [35]. The equations of the velocity and position vector adjustment of the particle in the iteration are as follows:

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 [P_{ib} - X_i(t)] + c_2 r_2 [P_G - X_i(t)], \quad (13)$$

$$X_i(t+1) = X_i(t) + V_i(t), \quad (14)$$

where the parameter  $\omega$  is the inertia weight; the parameters  $c_1$  and  $c_2$  are the learning factors, which represent the self-learning ability of the particle and the ability of social learning. The random acceleration factor pulls the particle toward the individual optimal  $P_{ib}$  and the group optimal  $P_G$ , and the parameters  $r_1$  and  $r_2$  are random numbers in the range of 0 to 1.

The speed is limited to the interval  $[-v_{\max}, v_{\max}]$  to limit the search of the particles to the optimal solution space. The value of  $v_{\max}$  plays a key role; if the value is very large, the particle will leave the solution space and cannot obtain the optimal solution. If the value of  $v_{\max}$  is very small, the particle easily falls into the local optimum, and it is difficult to obtain the global optimum solution. An appropriate value of the inertia weight  $\omega$  balances the global search and the local search and minimizes the number of iterations to obtain the optimal solution.

**3.3. Extreme Learning Machine Optimized by PSO Algorithm.** For the random selection of the parameters of the ELM, the traditional optimization only focuses on minimizing the RMSE and does not consider the output weight norm and the hidden layer output of the matrix condition number, which causes the network to crash [36, 37]. Therefore, in this study, we use the PSO to optimize the ELM (referred to as PSO-ELM). The implementation of the PSO-ELM algorithm is as follows:

- (1) The determination of the fitness function. First, the data set is divided into a training set, check set, and test set. In general, the RMSE is used as the fitness function. The samples of the check set are defined as  $n_v$ , and the RMSE is minimized during the iteration process. The RMSE is defined as follows

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^{n_v} \left\| \sum_{i=1}^H \beta_i g(\omega_i \cdot x_j + b_i) - t_j \right\|_2^2}{n_v}} \quad (15)$$

The 2-norm condition number (COND) is used to determine the condition of the matrix. In the limit learning

machine, the COND of the output matrix  $H$  of the hidden layer is calculated as follows:

$$\kappa_2(H) = \sqrt{\frac{\lambda_{\max}(H^T H)}{\lambda_{\min}(H^T H)}}, \quad (16)$$

where  $\lambda_{\max}(H^T H)$  and  $\lambda_{\min}(H^T H)$  are the maximum and minimum eigenvalues of the matrix  $H^T H$ , respectively. The closer the condition number  $k_2(H)$  of the 2-norm is to 1, the healthier the system is, and the larger the  $k_2(H)$  value is, the less healthy the system is, and the lower the generalization performance is. Therefore, the COND  $k_2(H)$  is used to measure the health of the system. The RMSE and the COND of the hidden layer output matrix  $H$  are selected as the fitness functions of the PSO. The COND and RMSE of the hidden layer output matrix  $H$  are obtained for the verification sample set. The fitness function is calculated as shown in Eq. (17):

$$f = \text{RMSE} + \text{COND}$$

$$= \sqrt{\frac{\sum_{j=1}^{n_v} \left\| \sum_{i=1}^H \beta_i g(\omega_i \cdot x_j + b_i) - t_j \right\|^2}{n_v}} + \kappa_2(H). \quad (17)$$

- (2) Update of the extreme value of the individual  $P_{ib}$  and the extreme value of the population  $P_G$ . The fitness value of each particle is determined according to the fitness function and is compared with the individual's board value and the population's extreme value in the current iteration. The particles with the smallest RMSE and COND represent the fitness values of the individual  $P_{ib}$  and those of the population  $P_G$ , as defined in Eqs. (18) and (19)

$$P_{ib} = \begin{cases} P_i (\text{RMSE}_{P_i} < \text{RMSE}_{P_{ib}}) \text{ and } (\text{COND}_{P_i} < \text{COND}_{P_{ib}}) \\ P_{ib}, \text{ else,} \end{cases} \quad (18)$$

$$P_G = \begin{cases} P_i (\text{RMSE}_{P_i} < \text{RMSE}_{P_G}) \text{ and } (\text{COND}_{P_i} < \text{COND}_{P_G}) \\ P_G, \text{ else,} \end{cases} \quad (19)$$

where  $\text{RMSE}_{P_i}$  is the RMSE of the  $i$ -th particle,  $\text{RMSE}_{P_{ib}}$  is the optimal RMSE of the  $i$ -th particle, and  $\text{RMSE}_{P_G}$  is the optimal RMSE of the particles. The parameter  $\text{COND}_{P_i}$  is the COND of the output matrix  $H$  of the hidden layer of the  $i$ -th particle,  $\text{COND}_{P_{ib}}$  is the COND of the output matrix  $H$  of the optimal hidden layer of the  $i$ -th particle, and  $\text{COND}_{P_G}$  is the COND of the optimal output matrix  $H$  of the hidden layer of the

particles. The flowchart of the PSO-ELM algorithm is shown in Figure 2 [36].

The steps of the PSO-ELM algorithm are as follows:

- (1) Random initialization of the particle swarms: The PSO algorithm is used to optimize the parameters of the ELM. Each particle  $P_i$  in the particle swarm consists of the ELM input weights  $\omega_j$  and bias  $b_j$ . The initial value of all the elements in the vector is a random number in the range  $[-1, 1]$
- (2) We calculate the fitness value of each particle using the fitness function, determine the training error, and output the weight. The training set and the check set in the iteration are randomly selected
- (3) We update the input weights and biases of the particles, train them with the training set, and obtain the error from the verification set. The information is used as the input layer of the ELM, and the adjustment values of the input weights and biases are calculated according to the error values. The velocity and position vectors of the particles are updated. The extreme values of the individual and the population are updated
- (4) After the iteration is completed, the fitness values of each particle are sorted, and the information of the  $M$  particles is selected to establish the ELM model
- (5) Using the test set, we extract the input weights and the offset values of the particles, and the combined prediction model of the ELM is established by using the output weights to test the test set
- (6) We use the test results of the combined model of the ELM and output the optimal results and the optimal PSO-ELM model

#### 4. Application of PSO-ELM Algorithm in Data Fusion of HWSNs

The materials and methods section should contain sufficient detail so that all procedures can be repeated. It may be divided into subsections if several methods are described.

*4.1. The Concept of the Proposed Algorithm.* In this study, data fusion is performed on the data of the cluster heads of the network layers and the members of the cluster nodes in an HWSN according to the spatiotemporal correlation of the data obtained by the sensor nodes.

The PSO-ELM data fusion algorithm is improved and optimized using the SEP clustering algorithm. The SEP algorithm clusters all the nodes in the detection area and randomly assigns values between 0 and 1. If it is less than a given threshold, the sensor node becomes a cluster head, and the cluster head and the cluster member node form a stable cluster structure. The member nodes in the cluster preprocess the collected data and transmit them to the cluster head. The cluster head fuses the data, removes the redundant information, and forwards the data to the

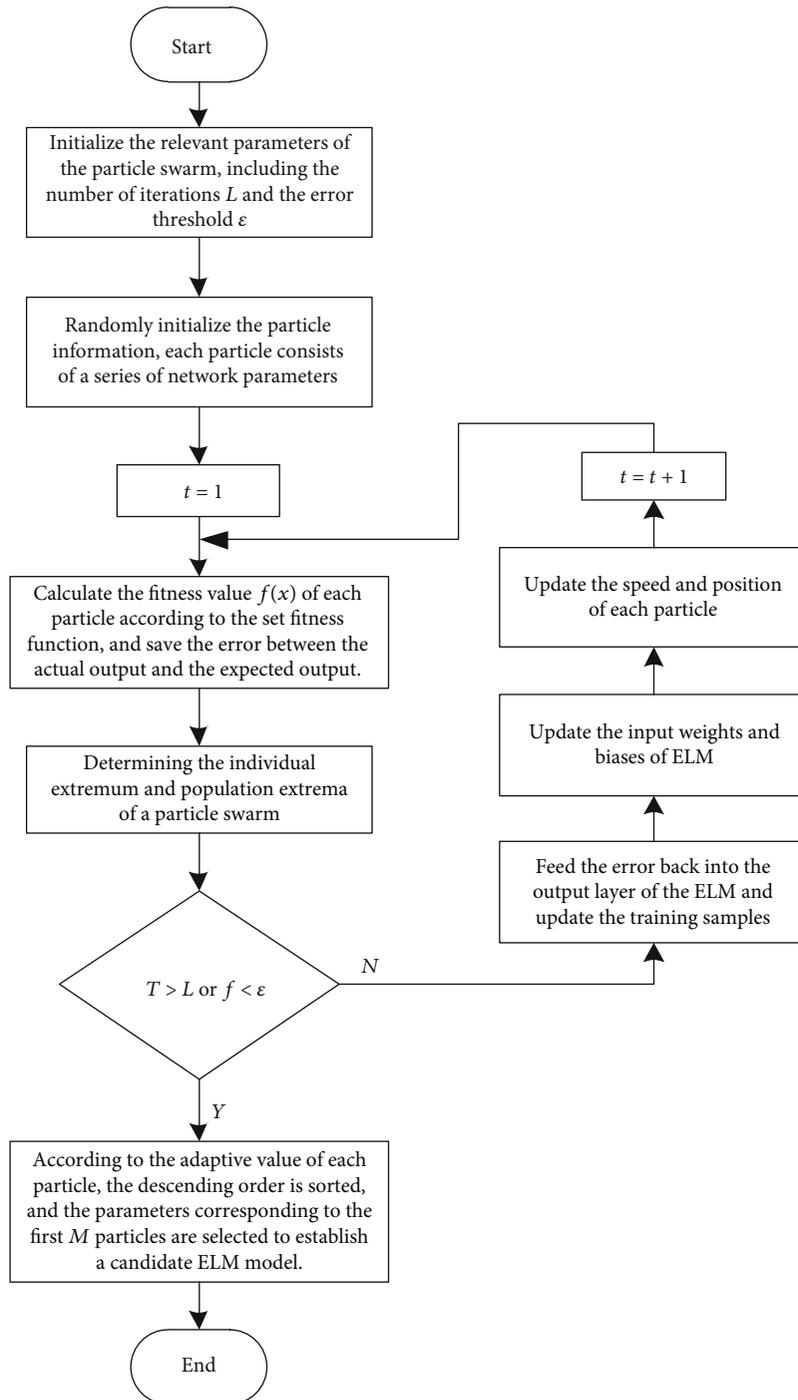


FIGURE 2: The flow chart of the PSO-ELM algorithm.

sink node. The PSO-ELM data fusion method processes the data between the member nodes and the cluster heads in the cluster.

**4.2. Implementation Steps of the Proposed Algorithm.** In existing clustering routing algorithms, it is necessary to reconstruct the clusters and change their topology in each round of loops, thereby increasing the network's energy requirement. In contrast, the proposed PSO-ELM data fusion algorithm ensures that the topology remains unchanged after

clustering the sensor nodes. When the cluster head is rotated, the sensor node with the highest energy is selected as the cluster head; this approach reduces the energy consumption during clustering.

In the PSO-ELM data fusion algorithm, the SEP routing protocol continuously adjusts the cluster head during the simulation, and the cluster reconstruction process is the process of regression. When the number of clusters of the HWSN has been stabilized, the fusion model of the PSO-ELM is trained. The cluster head node fits the original data

collected by the sensor nodes in the cluster, and parameter optimization is performed to improve the training speed of the network. Finally, the fusion results are transmitted to the sink node. A differential code division multiple access (CDMA) scheme is used for communication between clusters. The PSO-ELM data fusion algorithm has three phases. The first phase is the construction of clusters in the SEP, the second phase is the data fusion in the cluster using the ELM, and the third phase is data transmission. The PSO-ELM data fusion algorithm reduces the energy consumption of the sensor nodes by reducing the amount of data transmitted to the sink nodes.

In the data fusion process, the sensor nodes in the HWSN are initialized to determine the status of the normal nodes, the routing nodes, and the cluster head nodes in the network. Subsequently, the clustering structure is established according to the location of the sensor nodes in the monitoring area, and the cluster head node is randomly selected from the cluster members; the cluster head node acquires all the information from the cluster member nodes. After the clustering process has stabilized and the routing table of the normal node to the sink node has been constructed, the cluster head node sends the information from the cluster member nodes to the sink.

After the network is stable, the data fusion model is trained to obtain the number of nodes, the weights, and the threshold parameters of the hidden layer of the PSO-ELM. Due to the limited energy of the ordinary sensor nodes of the HWSN, data have to be obtained, and sending/receiving tasks have to be performed. The sink node constructs an extreme learning machine optimized by particle swarm optimization network architecture according to the received information. Samples are obtained that match the information of the member nodes in the sample database of the cluster head for training. After the training is completed, the sink node sends the parameters of the data fusion model (the number of hidden layer nodes, the network weights, and the thresholds) to the corresponding cluster head nodes, which fuse the data sent by the cluster member nodes, extract the features, delete redundant information, temporarily store the fused data, and then send the data to the sink node. The flowchart of the PSO-ELM data fusion algorithm is shown in Figure 3 [37], and the implementation steps are shown in Table 1.

## 5. Simulation Comparison and Performance Analysis

**5.1. Simulation Environment.** We use MATLAB 2014b simulation software to assess the performance of the proposed data fusion algorithm and compare it with that of other algorithms. We assume that the sensor nodes are randomly and evenly distributed in a two-dimensional space of  $500 \times 500 \text{ m}^2$ , and the number of sensor nodes is 200. The number of simulation rounds is 400. The sensor nodes send the data packets from the source node to the sink node; the packet has a capacity of 4 kb. The limits of the ELM parameters are as follows: the number of the initial hidden layer nodes of the network is 10, and we increase the number of

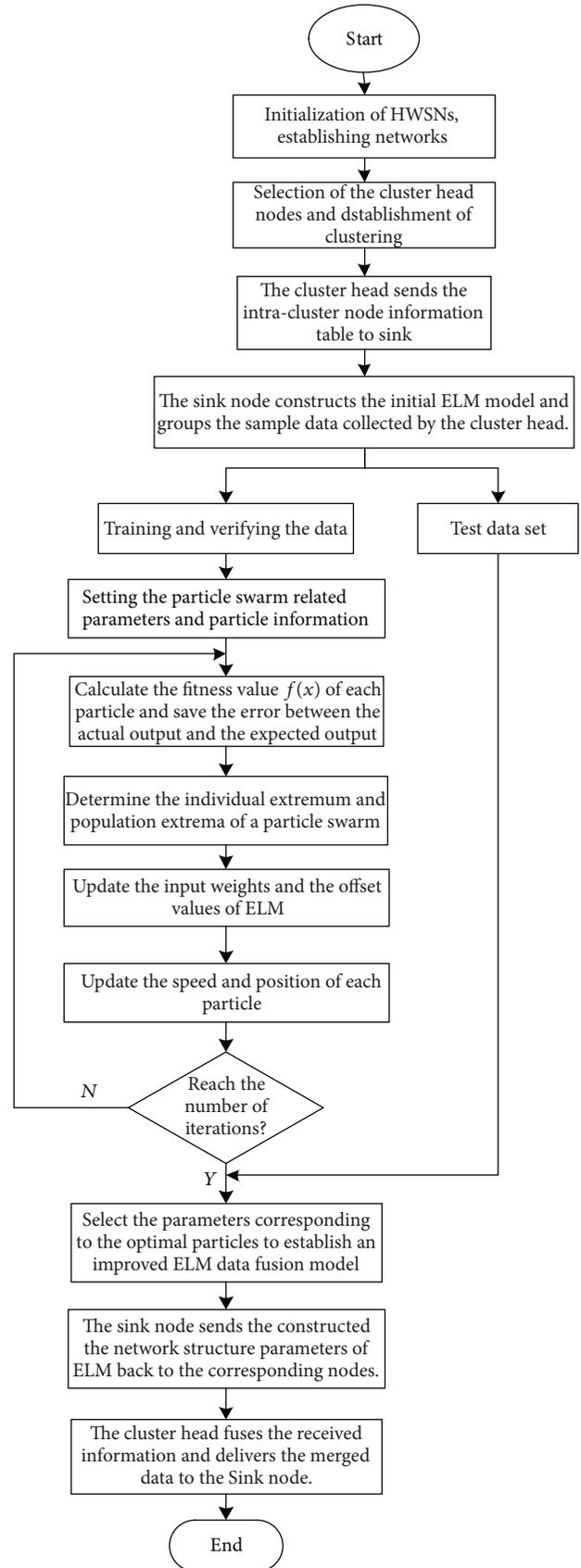


FIGURE 3: The data fusion algorithm of HWSNs based on PSO-ELM.

TABLE 1: Implementation steps of the data fusion algorithm in HWSNs.

Algorithm. Data fusion of HWSNs based on PSO-ELM.
Step 1: initialization of the parameters of HWSNs, such as the security verification of the sensor nodes, initialization of the network, location of sensing nodes, and the number of ELM hidden layer nodes, meanwhile, the parameters such as population space, inertia weight, and acceleration factor in the particle swarm optimization process are set, and the particle search space is defined.
Step 2: the data fusion model is initialized under the input weight and threshold constraints.
Step 3: for each particle in the population, the output weight matrix and output of extreme learning machine optimized by particle swarm optimization are calculated.
Step 3: the data fusion model is mapped to the PSO-ELM method, and the calculated the expected error is used as an optimization function of data fusion.
Step 4: the cluster head nodes are selected from the common nodes and the cluster head perception regions are established.
Step 5: the cluster head node sends the cluster member node information table to the sink node.
Step 6: the sink node constructs a PSO-ELM data fusion model and trains the sample data to obtain the extreme learning machine network parameters (weights and thresholds).
Step 7: if the termination condition of the network training setting is not reached, continue execution.
Step 8: the sink node sends the trained PSO-ELM network structure parameters to the corresponding nodes.
Step 9: the cluster head node performs data fusion on the received data by the trained PSO-ELM network structure. The fused data is then sent to the sink node.
Step 10: the data fusion algorithm for HWSNs is over.

hidden layer nodes by 20 cycles until the maximum value of 200 is reached. The hardlim learning function is used for the ELM's hidden layer activation function. The parameters of the PSO algorithm are as follows: the learning factors  $c_1$  and  $c_2$  of the PSO algorithm are 2.4, the initial inertia value  $\omega_{\max}$  is 0.9, the inertia weight  $\omega_{\min}$  for the iteration to the maximum algebraic expression is 0.2, the initial particle number is 30, and the number of iterations is 30. The results are the average values of 100 experiments. The parameter settings for the simulation are shown in Table 2.

We compare the performances of four data fusion algorithms, namely, the SEP algorithm [38], the backpropagation (BP) neural network [39], the ELM data fusion method [40], and the PSO-ELM method.

## 5.2. Algorithm Comparison and Result Analysis

### 5.2.1. Comparison of the Network Energy Consumption.

The energy consumption of a heterogeneous sensor network is defined as the sum of the energy consumed by the sensor nodes in the network. The total energy consumption of the four algorithms is shown in Figure 4.

As the number of simulation rounds increases, the average total energy consumption of the network increases for

TABLE 2: Simulation environment parameter settings.

Parameter	Numerical value
Network size	$500 \times 500 \text{ m}^2$
Number of nodes	200
Communication radius	80 m
Initial energy	1 J
$E_{\text{elec}}$	50 nJ/bit
$E_{fs}$	10 pJ/bit/m2
$E_{mp}$	0.0013 pJ/bit/m4
$l$	4000 bits
$d_0$	$\sqrt{E_{fs}/E_{mp}} = 87 \text{ m}$

the four algorithms. The total energy consumption is the largest after 200 rounds due to the large energy consumption of the nodes and the death of some nodes. The SEP algorithm has the highest growth rate and results in the highest energy consumption. The average energy consumption of the nodes in heterogeneous sensing network data fusion based on BP neural network increases greatly. The average energy consumption of the sensor nodes in the data fusion method based on extreme learning machine is small. In this paper, the data fusion method of HWSNs based on particle swarm optimization (PSO-ELM) is proposed to minimize the average energy consumption of nodes. For example, in the first 200 rounds, the proposed PSO-ELM method results in 71.4%, 39.5%, and 26.3% less energy consumption than the SEP algorithm, the BP neural network, and the ELM-based data fusion method, respectively.

### 5.2.2. Comparison of Node Survival.

The number of surviving sensor nodes is an indicator of the lifetime of HWSNs. The simulation time is based on the number of rounds of data transmission. The number of surviving nodes in the heterogeneous sensor network decreases with an increase in the number of rounds in the network simulation for all four algorithms, as shown in Figure 5.

At 100 rounds, the number of surviving network nodes of the heterogeneous sensing network data fusion SEP algorithm is greatly degraded. The BP data fusion algorithm has the lowest number of surviving nodes at 130 rounds, and the ELM data fusion algorithm has the lowest number of surviving nodes at 190 rounds. The lowest number of surviving nodes of the PSO-ELM algorithm occurs at 300 rounds. After 400 rounds, the number of surviving network nodes is zero for the four algorithms, and the simulation ends. The results show that the node survival time is longest for the proposed PSO-ELM algorithm.

### 5.2.3. Comparison of the Number of Cluster Head Nodes.

Generally, the optimal number of cluster head nodes is 6-10% of the total number of sensor nodes. The larger the number of cluster heads, the greater the network performance is. The number of cluster head nodes of the

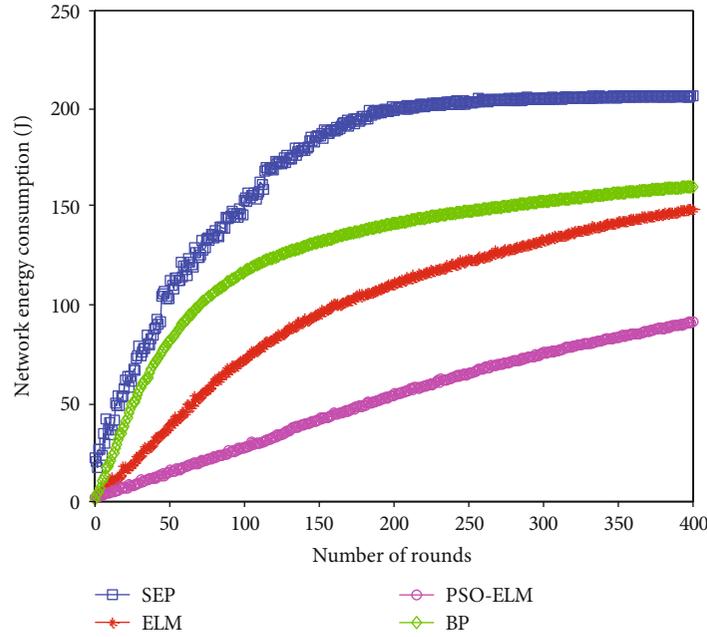


FIGURE 4: Comparison of the energy consumption.

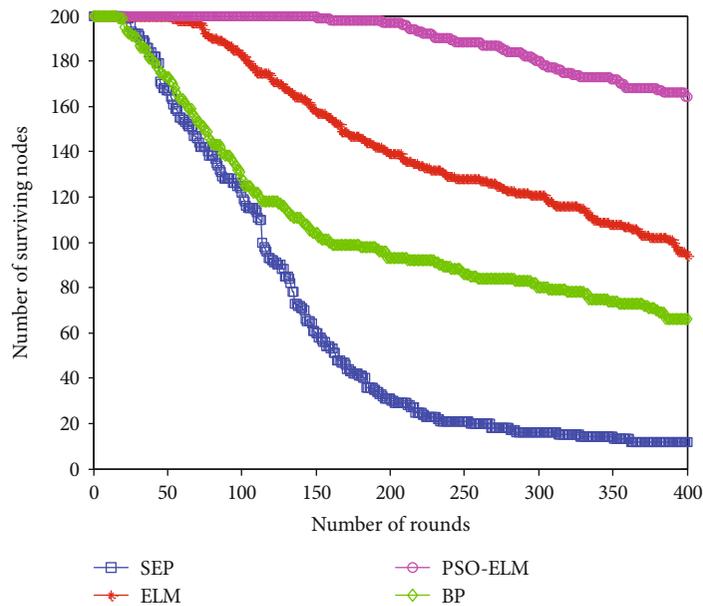


FIGURE 5: Comparison of the surviving sensor nodes.

heterogeneous sensor network for the four algorithms are shown in Figure 6.

The number of cluster head nodes is 20-28 for the PSO-ELM algorithm, 10-40 with large fluctuations for the SEP algorithm, 10-30 for the BP neural network, which has only 15 nodes in the cluster head, and 10-25 with large fluctuations for the ELM algorithm. The proposed algorithm performs best and has the lowest fluctuations.

5.2.4. Comparison of Energy Consumption of the Cluster Head Nodes. The energy consumption of the cluster head nodes is

an important index of network performance. The energy consumption of the cluster head nodes of the network for the four algorithms is shown in Figure 7.

The largest energy consumption is observed for the SEP algorithm, with an average of more than 10 mJ, followed by the BP neural network (7 mJ), the ELM algorithm (3 mJ), and the PSO-ELM algorithm (2 mJ). In addition, it can be seen that the proposed PSO-ELM algorithm consumes more energy from the cluster head nodes of the heterogeneous sensor network. The energy consumption of the cluster head nodes exhibits large fluctuations for the other three

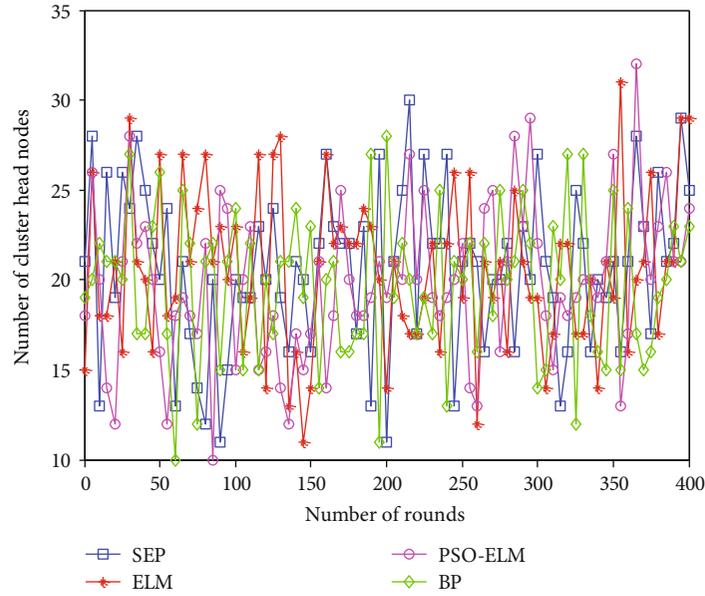


FIGURE 6: Comparison of the number of cluster head nodes.

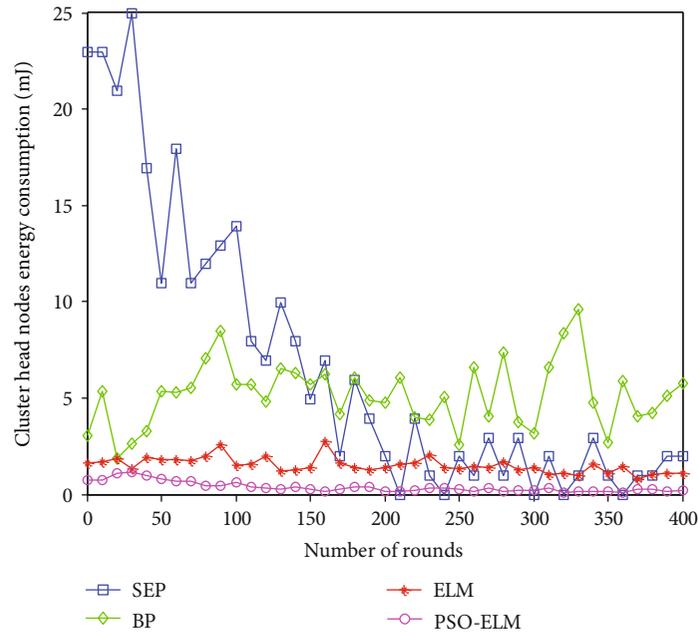


FIGURE 7: Energy consumption comparison of cluster head nodes.

algorithms. The proposed PSO-ELM method results in 80%, 71.4%, and 33.3% less energy consumption than the SEP algorithm, BP neural network algorithm, and ELM algorithm, respectively.

5.2.5. Comparison of Network Load Balancing. Load balancing is an important indicator of the network’s lifetime. The load balancing factor (LBF) is a convenient and effective indicator of network performance. The parameter  $L_{LBF}$  is defined as the reciprocal of the variance of the number of member-

aware nodes (including cluster head nodes) in a heterogeneous sensing network; it is calculated as follows:

$$L_{LBF} = \frac{n_c}{\sum_{i=1}^{n_c} (x_i - m)^2}, \tag{20}$$

where  $n_c$  is the number of nodes in the wireless sensor network;  $x_i$  is the number of sensing nodes in the  $i$ -th cluster head member;  $m$  is the average number of nodes in the

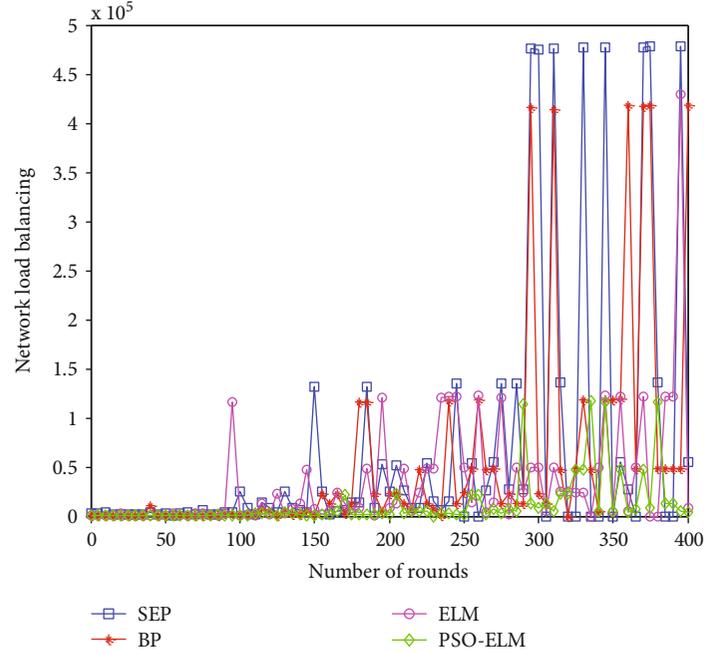


FIGURE 8: Comparison of network load balancing.

cluster head nodes. The LBF of the HWSN for the four algorithms is shown in Figure 8.

As shown in Figure 8, with the increase in the number of simulation rounds, the load balancing of the heterogeneous sensor network for the SEP algorithm is poor and fluctuates randomly. The BP neural network algorithm also results in poor load balancing and large random fluctuations. Better load balancing is observed for the ELM. The PSO-ELM algorithm provides the best load balancing performance; the longer the simulation time, the better the performance is.

**5.2.6. Comparison of Network Connectivity.** The continuous discretization method is generally used to calculate the connectivity of networks. For an HWSN, the node traversal method is used to calculate the connectivity at a specific state. An initial node is selected, and the perceptual node is used as a reference. The nodes that are directly connected, two-hop connected, and three-hop connected to the initial node are searched sequentially until the number of connected nodes no longer increased. The equation for calculating the connectivity rate is as follows:

$$N_{\text{con}} = \frac{N_l}{n}, \quad (21)$$

where  $N_l$  is the number of neighboring nodes in the communication range, and  $n$  is the number of all nodes in the network. The network connectivity for the four algorithms is shown in Figure 9.

As the number of simulation rounds increases, the network connectivity of the based on the SEP algorithm gradually decreases, and there are large fluctuations; the values range from 0.55 to 0.99. It is 0.73. The network connectivity based on the BP neural network shows large fluctuations, and the values range from 0.67 to 0.99, with an average of 0.81. For the ELM algorithm, the range is from 0.75 to 0.99,

stable, and the fluctuation range is large (0.3-1). The network connectivity of the ELM data fusion algorithm is relatively high, stable, and the fluctuation range is between 0.5 and 1. The proposed PSO-ELM algorithm has the highest network connectivity and overall stability. The range of network connectivity is between 0.82 and 1, and the network connectivity performance is excellent.

**5.2.7. Comparison of Network Reliability.** The overall network reliability  $R_{\text{net}}$  consists of the connectivity of the network  $C_1$ , the connectivity rate of the network  $C_2$ , and the capacity of the network  $C_3$ . Its calculation formula is

$$R_{\text{net}} = 0.1667C_1 + 0.5C_2 + 0.3333C_3 \quad (22)$$

The network node connectivity reliability  $C_1$  refers to the reliability of the connectivity of the end-to-end nodes. A reliability matrix is generally calculated based on the distance between the sensing nodes in the HWSNs and is compared with a reliability matrix obtained from random samples. A Monte Carlo analysis is conducted, and the average network reliability value after 50 rounds is obtained. The capacity of the network  $C_3$  is the network's survival probability. Generally, the number of surviving nodes of the network is divided by the total number of nodes to obtain the network's survival probability. The network reliability results for the four algorithms are shown in Figure 10.

As the number of simulation rounds increases, the network's overall reliability based on the SEP algorithm gradually decreases, and there are large fluctuations; the values range from 0.55 to 0.99. It is 0.73. The network reliability based on the BP neural network shows large fluctuations, and the values range from 0.67 to 0.99, with an average of 0.81. For the ELM algorithm, the range is from 0.75 to 0.99,

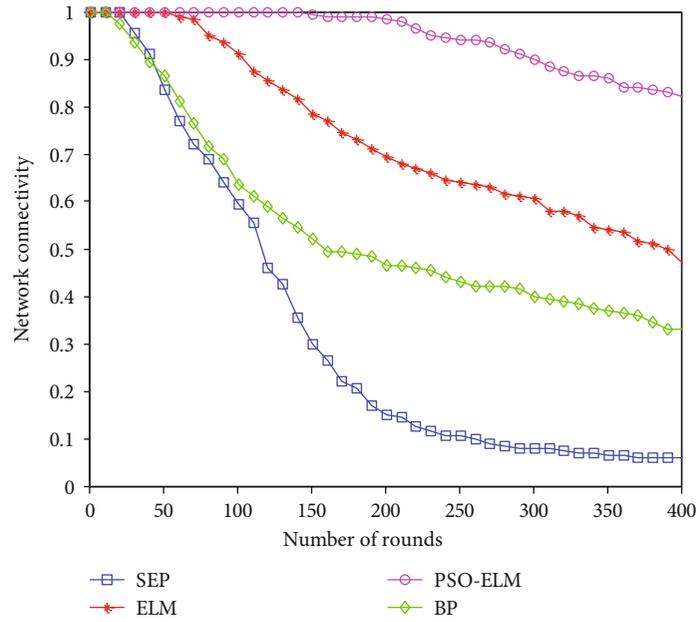


FIGURE 9: Comparison of network connectivity.

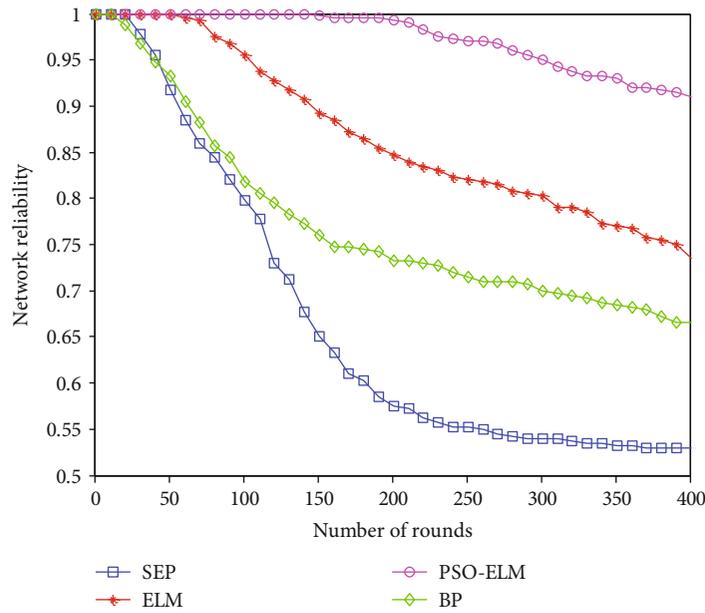


FIGURE 10: Comparison of the reliability of the four algorithms.

with an average of 0.88. The PSO-ELM algorithm has the smallest range of reliability (0.92 to 0.99), with an average of 0.97.

5.2.8. Comparison of 3D Graphs of the Average Energy Consumption. The 3D graphs of the average energy consumption of the network nodes based on the four algorithms are shown in Figure 11.

The average energy consumption of the network nodes is highest for the SEP algorithm (0.51 J), followed by the BP neural network (0.38 J), the ELM data fusion algorithm (0.26 J), and the PSO-ELM algorithm (0.11 J).

The simulation results show that the performance is highest for the proposed PSO-ELM data fusion algorithm. The SEP algorithm is a typical routing protocol for heterogeneous wireless sensor networks. It does not use the data fusion methods. The fusion speed based BP neural network method is not fast enough, and the accuracy rate is not accurate enough. The data fusion based ELM method has insufficient weight and threshold optimization, and the fusion accuracy is not accurate. The proposed PSO-ELM method uses the particle swarm optimization algorithm to optimize the weights and thresholds of the extreme learning machine, improve the fusion accuracy of the

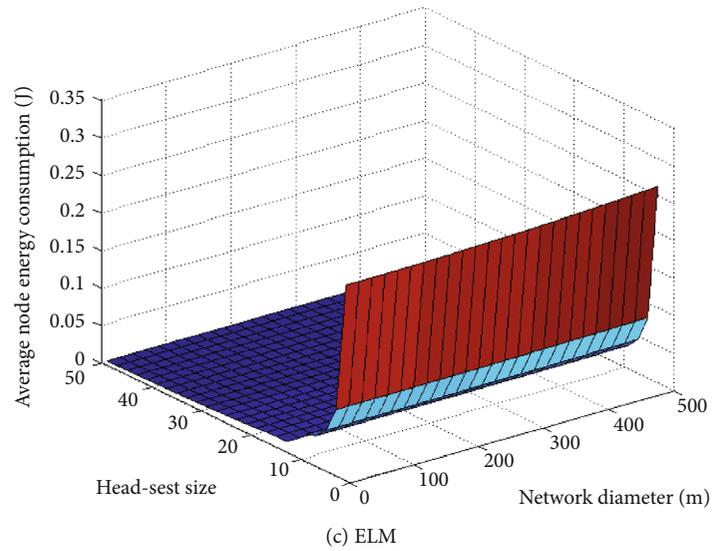
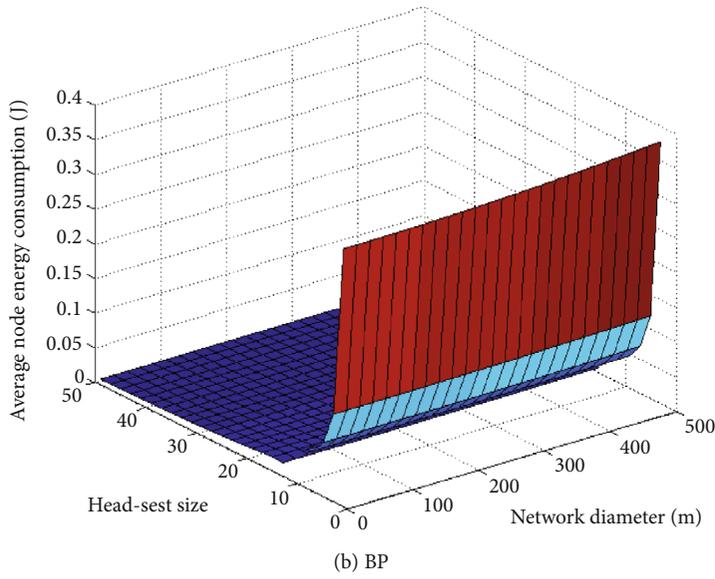
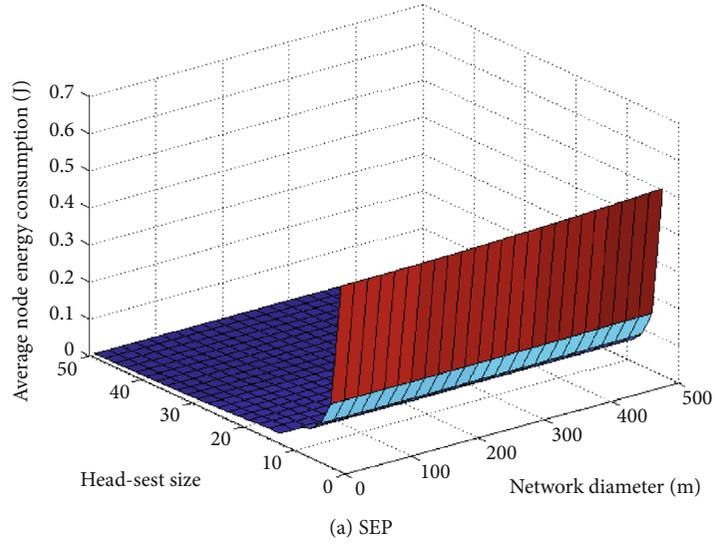


FIGURE 11: Continued.

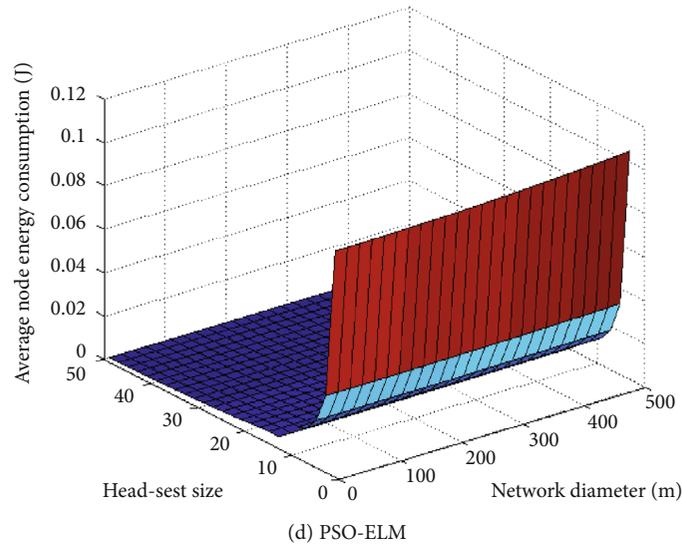


FIGURE 11: 3D comparison of the average energy consumption.

algorithm, improve the network's lifetime, and reduce the network energy consumption. These results are consistent with the simulation results of the other evaluation indices. The PSO-ELM method has the highest efficiency, lowest energy consumption, and the highest reliability for data fusion in HWSNs.

## 6. Conclusions

A data fusion method for HWSNs based on the PSO-ELM algorithm was proposed. The spatiotemporal correlation between the data obtained by the sensor nodes of the HWSNs was determined, and the ELM was used to process the data collected by the sensor nodes in the hierarchical routing structure of the HWSNs. The PSO-ELM method resulted in a well-balanced energy load of the network, a reduction in energy consumption, thereby increasing the lifetime of the network.

In future research, we will focus on three aspects: (1) We plan to integrate the sensor node distance and residual energy of the HWSN to improve and optimize the ELM, simplify the data fusion process to improve the network performance, and conduct an in-depth study of the complex calculations of intelligent optimization algorithms. (2) The addition of new nodes after the death of a node maintains the communication capability of the network. The strategy of the dynamic addition of new nodes will be investigated, and other data fusion algorithms for HWSNs will be explored, such as the P-SEP algorithm. (3) We will focus on data fusion in large-scale HWSNs; since the base station and the cluster heads are far apart, it is necessary to use the inter-cluster multihop method to solve for long-distance data transmission, and multiple mobile sink nodes are needed for data fusion and transmission. These future research directions are important topics for studies on data fusion in HWSNs.

## Data Availability

No specific dataset is required.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the National Nature Science Foundation of China under Grant 61201247 and 61801319, in part by the Science and Technology Development Program of Central Guide to Local Government of China under Grant 2019ZYDD043, in part by the International Science & Technology Cooperation Program of Hubei Province under Grant 2019AHB059, Project of Natural Science Foundation of Hubei Province, and the Talent Introduction Project of Hubei University of Arts and Science (2059072).

## References

- [1] T. Li, M. Zhao, and K. K. L. Wong, "Machine learning based code dissemination by selection of reliability mobile vehicles in 5G networks," *Computer Communications*, vol. 152, no. 2, pp. 109–118, 2020.
- [2] R. Amin, S. K. H. Islam, N. Kumar, and K.-K. R. Choo, "An untraceable and anonymous password authentication protocol for heterogeneous wireless sensor networks," *Journal of Network and Computer Applications*, vol. 104, no. 2, pp. 133–144, 2018.
- [3] M. Chen, T. Wang, K. Ota, M. Dong, M. Zhao, and A. Liu, "Intelligent resource allocation management for vehicles network: An A3C learning approach," *Computer Communications*, vol. 151, no. 2, pp. 485–494, 2020.
- [4] L. Cao, Y. Cai, and Y. Yue, "Swarm intelligence-based performance optimization for mobile wireless sensor networks:

- survey, challenges, and future directions,” *IEEE Access*, vol. 7, no. 7, pp. 161524–161553, 2019.
- [5] Y. G. Yue and P. He, “A comprehensive survey on the reliability of mobile wireless sensor networks: taxonomy, challenges, and future directions,” *Information Fusion*, vol. 44, pp. 188–204, 2018.
  - [6] S. Dutt, S. Agrawal, and R. Vig, “Cluster-head restricted energy efficient protocol (CREEP) for routing in heterogeneous wireless sensor networks,” *Wireless Personal Communications*, vol. 100, no. 4, pp. 1477–1497, 2018.
  - [7] W. Osamy, A. Salim, and A. M. Khedr, “An information entropy based-clustering algorithm for heterogeneous wireless sensor networks,” *Wireless Networks*, vol. 26, no. 3, pp. 1869–1886, 2020.
  - [8] R. M. Al-Kiyumi, C. H. Foh, S. Vural, P. Chatzimisios, and R. Tafazolli, “Fuzzy logic-based routing algorithm for lifetime enhancement in heterogeneous wireless sensor networks,” *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 2, pp. 517–532, 2018.
  - [9] X. Deng, Y. Jiang, L. T. Yang, M. Lin, L. Yi, and M. Wang, “Data fusion based coverage optimization in heterogeneous sensor networks: a survey,” *Information Fusion*, vol. 52, no. 12, pp. 90–105, 2019.
  - [10] X. Bai, Z. Wang, L. Sheng, and Z. Wang, “Reliable data fusion of hierarchical wireless sensor networks with asynchronous measurement for greenhouse monitoring,” *IEEE Transactions on Control Systems Technology*, vol. 27, no. 3, pp. 1036–1046, 2019.
  - [11] T. Wang, D. Zhao, S. Cai, W. Jia, and A. Liu, “Bidirectional prediction-based underwater data collection protocol for end-edge-cloud orchestrated system,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4791–4799, 2020.
  - [12] G. Yogarajan and T. Revathi, “Improved cluster based data gathering using ant lion optimization in wireless sensor networks,” *Wireless Personal Communications*, vol. 98, no. 3, pp. 2711–2731, 2018.
  - [13] S. Din, A. Ahmad, A. Paul, M. M. Ullah Rathore, and G. Jeon, “A cluster-based data fusion technique to analyze big data in wireless multi-sensor system,” *IEEE Access*, vol. 5, no. 1, pp. 5069–5083, 2017.
  - [14] S. Ji, C. Tan, P. Yang, Y. J. Sun, D. Fu, and J. Wang, “Compressive sampling and data fusion-based structural damage monitoring in wireless sensor network,” *The Journal of Supercomputing*, vol. 74, no. 3, pp. 1108–1131, 2018.
  - [15] M. Dong, K. Ota, and A. Liu, “RMER: reliable and energy-efficient data collection for large-scale wireless sensor networks,” *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 511–519, 2016.
  - [16] M. Wu, L. Tan, and N. Xiong, “Data prediction, compression, and recovery in clustered wireless sensor networks for environmental monitoring applications,” *Information Sciences*, vol. 329, no. 2, pp. 800–818, 2016.
  - [17] K. Xiao, J. Li, and C. Yang, “Exploiting correlation for confident sensing in fusion-based wireless sensor networks,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 6, pp. 4962–4972, 2018.
  - [18] L. A. Villas, A. Boukerche, H. A. B. F. de Oliveira, R. B. de Araujo, and A. A. F. Loureiro, “A spatial correlation aware algorithm to perform efficient data collection in wireless sensor networks,” *Ad Hoc Networks*, vol. 12, no. 1, pp. 69–85, 2014.
  - [19] S. B. Pourpeighambar and M. Sabaei, “Spatial correlation aware protocols for efficient data aggregation of moving object in wireless sensor networks,” *Scientia Iranica*, vol. 20, no. 3, pp. 695–709, 2012.
  - [20] L. A. Villas, A. Boukerche, D. L. Guidoni, H. A. B. F. de Oliveira, R. B. de Araujo, and A. A. F. Loureiro, “An energy-aware spatio-temporal correlation mechanism to perform efficient data collection in wireless sensor networks,” *Computer Communications*, vol. 36, no. 9, pp. 1054–1066, 2013.
  - [21] R. G. Cid-Fuentes, A. Cabellos-Aparicio, and E. Alarcon, “Energy buffer dimensioning through energy-erlangs in spatio-temporal-correlated energy-harvesting-enabled wireless sensor networks,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 4, no. 3, pp. 301–312, 2014.
  - [22] M. Arunraja and V. Malathi, “Collective prediction exploiting spatio temporal correlation (CoPeST) for energy efficient wireless sensor networks,” *KSII Transactions on Internet and Information Systems*, vol. 9, no. 7, pp. 2488–2511, 2015.
  - [23] X. Piao, Y. Hu, Y. Sun, B. Yin, and J. Gao, “Correlated spatio-temporal data collection in wireless sensor networks based on low rank matrix approximation and optimized node sampling,” *Sensors*, vol. 14, no. 12, pp. 23137–23158, 2014.
  - [24] S. Das, S. Barani, S. Wagh, and S. S. Sonavane, “Extending lifetime of wireless sensor networks using multi-sensor data fusion,” *Sādhanā*, vol. 42, no. 7, pp. 1083–1090, 2017.
  - [25] J. Rejina Parvin and C. Vasanthanayaki, “Particle swarm optimization-based clustering by preventing residual nodes in wireless sensor networks,” *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4264–4274, 2015.
  - [26] S. Rodríguez, J. F. de Paz, G. Villarrubia, C. Zato, J. Bajo, and J. M. Corchado, “Multi-agent information fusion system to manage data from a WSN in a residential home,” *Information Fusion*, vol. 23, no. 1, pp. 43–57, 2015.
  - [27] G. B. Huang, Q. Y. Zhu, and C. K. Siew, “Extreme learning machine: theory and applications,” *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
  - [28] S. Ding, H. Zhao, Y. Zhang, X. Xu, and R. Nie, “Extreme learning machine: algorithm, theory and applications,” *Artificial Intelligence Review*, vol. 44, no. 1, pp. 103–115, 2015.
  - [29] J. Tang, C. Deng, and G. B. Huang, “Extreme learning machine for multilayer perceptron,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 809–821, 2016.
  - [30] G. B. Huang, Z. Bai, L. L. C. Kasun, and C. M. Vong, “Local receptive fields based extreme learning machine,” *IEEE Computational Intelligence Magazine*, vol. 10, no. 2, pp. 18–29, 2015.
  - [31] L. Cao, Y. Cai, Y. Yue, S. Cai, and B. Hang, “A novel data fusion strategy based on extreme learning machine optimized by bat algorithm for mobile heterogeneous wireless sensor networks,” *IEEE Access*, vol. 8, no. 1, pp. 16057–16072, 2020.
  - [32] R. Cheng and Y. Jin, “A social learning particle swarm optimization algorithm for scalable optimization,” *Information Sciences*, vol. 291, pp. 43–60, 2015.
  - [33] P. Ghamisi and J. A. Benediktsson, “Feature selection based on hybridization of genetic algorithm and particle swarm optimization,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 2, pp. 309–313, 2015.
  - [34] Y. J. Gong, J. J. Li, Y. Zhou et al., “Genetic learning particle swarm optimization,” *IEEE Transactions on Cybernetics*, vol. 46, no. 10, pp. 2277–2290, 2016.
  - [35] M. Taherkhani and R. Safabakhsh, “A novel stability-based adaptive inertia weight for particle swarm optimization,” *Applied Soft Computing*, vol. 38, pp. 281–295, 2016.

- [36] N. Zeng, H. Zhang, W. Liu, J. Liang, and F. E. Alsaadi, "A switching delayed PSO optimized extreme learning machine for short-term load forecasting," *Neurocomputing*, vol. 240, no. 1, pp. 175–182, 2017.
- [37] H. Lu, B. du, J. Liu, H. Xia, and W. K. Yeap, "A kernel extreme learning machine algorithm based on improved particle swarm optimization," *Memetic Computing*, vol. 9, no. 2, pp. 121–128, 2017.
- [38] P. G. V. Naranjo, M. Shojafar, H. Mostafaei, Z. Pooranian, and E. Baccarelli, "P-SEP: a prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks," *The Journal of Supercomputing*, vol. 73, no. 2, pp. 733–755, 2017.
- [39] T. Wang and Y. Xie, "BP-GA data fusion algorithm studies oriented to smart home," *Mathematical Modelling of Engineering Problems*, vol. 3, no. 3, pp. 135–140, 2016.
- [40] Y. Yue, J. Li, H. Fan, and Q. Qin, "Optimization-based extreme learning machine for data fusion in mobile wireless sensor networks," *International Journal of Innovative Computing, Information and Control*, vol. 12, no. 5, pp. 1423–1438, 2016.