

Research Article

A Hybrid Model of Extreme Learning Machine Based on Bat and Cuckoo Search Algorithm for Regression and Multiclass Classification

Qinwei Fan  and **Tongke Fan**

Engineering College, Xi'an International University, Xi'an 710077, China

Correspondence should be addressed to Qinwei Fan; qinweifan@126.com

Received 21 September 2021; Accepted 11 October 2021; Published 8 November 2021

Academic Editor: Sun Young Cho

Copyright © 2021 Qinwei Fan and Tongke Fan. 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.

Extreme learning machine (ELM), as a new simple feedforward neural network learning algorithm, has been extensively used in practical applications because of its good generalization performance and fast learning speed. However, the standard ELM requires more hidden nodes in the application due to the random assignment of hidden layer parameters, which in turn has disadvantages such as poorly hidden layer sparsity, low adjustment ability, and complex network structure. In this paper, we propose a hybrid ELM algorithm based on the bat and cuckoo search algorithm to optimize the input weight and threshold of the ELM algorithm. We test the numerical experimental performance of function approximation and classification problems under a few benchmark datasets; simulation results show that the proposed algorithm can obtain significantly better prediction accuracy compared to similar algorithms.

1. Introduction

In recent years, artificial intelligence algorithms have drawn extensive attention from scientific research. As an important part of artificial intelligence, machine learning has been widely used in data mining [1], speech recognition [2], feature selection [3, 4], learning incentivization strategy [5], natural language processing [6], and the nonlinear function approximation and benchmark problem [7]. As a branch of machine learning, neural networks have been successfully applied in many tasks of learning from data. However, most of the traditional neural networks use the gradient learning algorithm for network training, which makes the network make problems such as low training efficiency, slow speed, and easy to fall into local optimal.

Extreme Learning Machine (ELM) is a new method of training artificial neural networks and includes supervised training methods, which is a kind of neural network structure put forward by Huang et al. using single hidden layer feedforward networks (SLFN) [8–10]. Huang et al. [11]

argue that the existing neural networks have some defects in learning speed; the main reason for the low rate of learning is that all the parameters on the network are determined repeatedly by a training method. In the ELM learning algorithm, the weight feedback and threshold are generated randomly. Then, the output of the hidden layer matrix is used to calculate the final output weight. Computing the final weights was obtained using Moore–Penrose (MP) generalized inverse. Compared with other neural networks based on the gradient learning algorithm, the ELM learning algorithm has great advantages in learning speed, and it is capable of producing good generalization performance and greatly reduces the computational complexity of complex application problems [12, 13]. Meanwhile, these good performances have been widely promoted in various practical application fields such as biomedicine [14–16], fault diagnosis [17, 18], and indoor positioning systems [19, 20]. However, since the input parameters are generated randomly and the ELM requires a large number of hidden neurons, the amplitude of the output weight will be large

when the output matrix of the hidden layer is ill, which will cause the trained model to fall into the local minimum and show the phenomenon of overfitting [21]. In [22, 23], an ELM based on different regularization was proposed to effectively overcome the overfitting phenomenon. The accuracy and effectiveness of the ELM algorithm largely rest with the internal parameters of the model. So as to choose the suitable model parameters, many researchers use a bionic optimization algorithm to optimize the input weights and thresholds.

In the literature [24], the improved ELM algorithm was proposed, which used a differential evolution algorithm to choose the input weights and then used MP generalized inverse analysis to determine the output weights. This improvement enables it to obtain better generalization performance in a compact network. In the literature [25], the coral reefs optimization (CRO – ELM) has been used for carrying out evolution in ELM weights to enhance the performance of these machines. A new evolutionary algorithm, particle swarm optimization (PSO – ELM), is introduced to optimize the input weight and hidden bias of ELM [26, 27] so that the network has better generalization performance in the benchmark classification experiment and is more suitable for some prediction problems. A real-coded genetic algorithm (RCGA – ELM) was proposed [28] to select the number of hidden neurons and the input weights, such that the generalization performance of the classifier is a maximum. But it needed to adjust many parameters in genetic operators artificially. The cuckoo search algorithm (CS – ELM) was proposed [29–33], which was used to pretrain the ELM ensuring optimal solutions and to further improve the accuracy and stability of CS – ELM. References [34, 35] proposed ICS model, which combines the improved cuckoo search algorithm with ELM. Both CS – ELM and ICS – ELM select the input weights and biases before calculating the output weights, and they ensure the full column rank of the hidden layer output matrix.

Bat algorithm (BA) [36, 37] and cuckoo search algorithm (CS) [38, 39] are two new heuristic swarm intelligence optimization algorithms. Bat algorithm has the advantages of a simple model, fast convergence rate, strong global optimization, and so on and has been widely used in engineering optimization, model identification, and other problems. The cuckoo search algorithm has the characteristics of simple and efficient, few parameters, easy to implement, and excellent random search path and has been successfully applied to medical image optimization [40], multiobjective optimization [41], image processing [42], and other practical problems. Literature [43] shows that bat algorithm and cuckoo search algorithm have great advantages over genetic algorithm and particle swarm optimization in the new metaheuristic environment. In this paper, we combine the BACS hybrid algorithm with traditional ELM and propose an optimization algorithm of ELM based on BACS. The basic thought of the BACS – ELM algorithm is to use the BACS algorithm to train the input weight and threshold value randomly generated by ELM to find the optimal parameter and then determine the output weights

by using MP generalized inverse so as to improve the convergence speed and stability of the network model. The main contributions are as follows:

- (1) Based on the idea of a group intelligence optimization algorithm, this paper introduces how to train ELM by BACS hybrid algorithm. By using this method, the input weights and thresholds of the ELM network can be reasonably optimized to solve the randomness problem of hidden layer parameters so that the network parameters can reach the optimum.
- (2) By improving the traditional ELM network by BACS hybrid algorithm, the local and global optimization problems are effectively balanced, and the generalization performance of the network is improved.
- (3) Nonlinear function fitting and classification problems present that the BACS – ELM algorithm can acquire better approximation effect and generalization performance than other algorithms.

The rest of the paper is arranged as follows: Section 2 introduces the traditional ELM network model and algorithm. Section 3 introduces the principles and implementation steps of the bat algorithm and cuckoo search algorithm. The hybrid algorithm of Extreme Learning Machine based on the bat cuckoo algorithm is described in Section 4. Some numerical experiments are discussed in Section 5. Section 6 offers some conclusions for this paper.

2. The Preliminary of ELM

In this section, we begin with the introduction of standard ELM, the network model of ELM is shown in Figure 1, and its network model can be divided into three layers, which are the input layer, hidden layer, and output layer. All of these works provide fundamental theoretical support for the new method proposed next. $(x_j, o_j) \in R^n \times R^m$ represents P arbitrary various samples, where $x_j = (x_{j1}, x_{j2}, \dots, x_{jm})^T \in R^n$ and $o_j = (o_{j1}, o_{j2}, \dots, o_{jm})^T \in R^m$; the traditional SLFN with L hidden nodes can be mathematically modeled as

$$h_L(\mathbf{x}_j) = \sum_{i=1}^L \beta_i G(\mathbf{w}_i, b_i, \mathbf{x}_j) = \mathbf{t}_j, \quad j = 1, 2, \dots, P, \quad (1)$$

where $G(\mathbf{w}_i, b_i, \mathbf{x}_j)$ is an activation function, which can take various kinds forms, such as the sigmoid function:

$$G(\mathbf{w}, b, \mathbf{x}) = \frac{1}{1 + \exp(-(\mathbf{w}^T \mathbf{x} + b))} \quad (2)$$

or Gaussian function:

$$G(\mathbf{w}, b, \mathbf{x}) = \exp(-b\|\mathbf{w} - \mathbf{x}\|^2). \quad (3)$$

The above SLFN can approximate these P samples in the training process of gradual iteration. When the learning error is reduced to zero, $\sum_{j=1}^P \|\mathbf{t}_j - \mathbf{o}_j\| = 0$, the learning capacity of the ELM is optimal, and then there exist (\mathbf{w}_i, b_i) and β_i such that

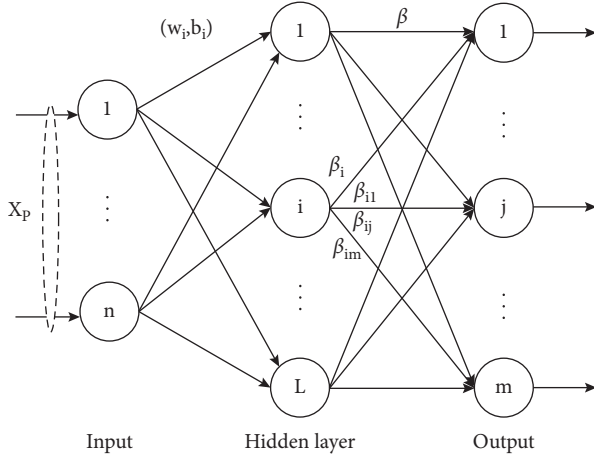


FIGURE 1: The structure of the ELM model.

$$\sum_{i=1}^L \beta_i G(\mathbf{w}_i, b_i, \mathbf{x}_j) = \mathbf{o}_j, \quad (4)$$

where $\mathbf{w}_i = (w_{i1}, w_{i2}, \dots, w_{in})^T \in \mathbf{R}^n$ is the input weight, which links the i -th hidden node as presented in Figure 1, $b_i \in \mathbf{R}$ is the threshold of the i -th hidden node and is generated randomly, $\beta_i = (\beta_{i1}, \beta_{i2}, \dots, \beta_{im})^T \in \mathbf{R}^m$ is the output weight of the i -th hidden node, and \mathbf{t}_j represents the actual output of input \mathbf{x}_j in the network.

The above P equations can be rewritten as the following matrix form:

$$\mathbf{H}\beta = \mathbf{O}, \quad (5)$$

where

$$\begin{aligned} \mathbf{H}(\mathbf{w}_1, \dots, \mathbf{w}_L, b_1, \dots, b_L, \mathbf{x}_1, \dots, \mathbf{x}_P) \\ = \begin{bmatrix} G(\mathbf{w}_1, b_1, \mathbf{x}_1) & \cdots & G(\mathbf{w}_L, b_L, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ G(\mathbf{w}_1, b_1, \mathbf{x}_P) & \cdots & G(\mathbf{w}_L, b_L, \mathbf{x}_P) \end{bmatrix}_{P \times L}, \\ \beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \\ \mathbf{O} = \begin{bmatrix} \mathbf{o}_1^T \\ \vdots \\ \mathbf{o}_P^T \end{bmatrix}_{P \times m}, \end{aligned} \quad (6)$$

where H is called the output matrix of the hidden layer and β represents the final output matrix. The basic principle of ELM is to obtain the output weight β through formula $\mathbf{H}\beta = \mathbf{O}$.

In practical training, the number of nodes L in the hidden layer is usually less than the number of training samples P . Therefore, on the premise that the activation function is differentiable, input weights and thresholds randomly selected before training should remain unchanged during training. In this way, the output weight of the

network can be obtained by solving the least squares of the following linear system:

$$\min_{\beta} \|\mathbf{H}\beta - \mathbf{O}\|, \quad (7)$$

and the explicit solution is

$$\hat{\beta} = \mathbf{H}^{\dagger} \mathbf{O}, \quad (8)$$

where \mathbf{H}^{\dagger} represents the MP generalized inverse of \mathbf{H} [44]. Therefore, ELM can be described as follows (Algorithm 1).

3. Algorithm Description

3.1. Bat Algorithm. Bat algorithm (BA) is a swarm intelligence optimization algorithm that simulates the predation behavior of bats. Because of its simple model, fast convergence speed, and strong global optimization, it has been widely used in data mining, wireless sensors, and power systems. However, there are also some problems in practical applications, such as easy premature convergence and low optimization accuracy.

The bat algorithm determines the optimal bat in the current search space by adjusting the frequency, wavelength, and loudness and then obtains the optimal solution to the optimization problem. For this algorithm, in order to simulate this predation behavior, the following assumptions are proposed in the process:

- (1) All bat individuals can use echolocation to perceive the distance and distinguish the difference between the target and the obstacle in a special way
- (2) The bat flies randomly at position x_i at speed v_i , finds the target with frequency f_{\min} , variable wavelength λ , and loudness A_0 , and automatically adjusts the wavelength (or frequency) and pulse emission rate $r \in [0, 1]$ through the distance from the target and so on
- (3) Assume that the loudness changes from the maximum value A_0 to the minimum value A_{\min}

Assuming that, in the search space with dimension d , the number of iterations is t , the update formulas for the frequency, velocity, and position of the bat individual i in the t -th generation are as follows:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta, \quad (9)$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x^*)f_i, \quad (10)$$

$$x_i^t = x_i^{t-1} + v_i^t, \quad (11)$$

where f_i represents the frequency of the i -th bat and its adjustment range is $[f_{\min}, f_{\max}]$, β is a random number that obeys a uniform distribution in $[0, 1]$, and x^* represents the current optimal solution.

For the current local search domain, a random number rand_1 is generated. If $\text{rand}_1 > \text{rand}_2$, the current new solution is generated by the random disturbance of the optimal solution. The update formula is as follows:

Input: given a training set $(\mathbf{x}_j, \mathbf{o}_j) \in \mathbf{R}^n \times \mathbf{R}^m$, activation function is $G(\mathbf{w}_i, b_i, \mathbf{x}_j)$, and the hidden nodes number is L .
Output β .
Step 1: setting learning parameters for hidden nodes \mathbf{w}_i and b_i , $1 \leq i \leq L$.
Step 2: calculate the output matrix \mathbf{H} based on (5).
Step 3: calculate the output weight $\beta = \mathbf{H}^+ \mathbf{O}$.

ALGORITHM 1: ELM algorithm.

$$x_{\text{new}} = x_{\text{old}} + \varepsilon A^t, \quad (12)$$

where ε is a random number in $[-1, 1]$ and A^t represents the average loudness of the bat population.

When the bat is constantly approaching the target, its loudness A will drop to a fixed value, and at this time, r will continue to increase. Randomly generate a number rand_2 ; if $\text{rand}_2 < A_i$ and the new fitness value $f(x_{\text{new}}) > f(x_{\text{old}})$, the new solution generated by (12) is accepted; that is $x_i^{t+1} = x_{\text{new}}$. The update formula for the loudness A_i and pulse rate r_i of the first bat is as follows:

$$A_i^{t+1} = \alpha A_i^t, \quad (13)$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\sigma t)], \quad (14)$$

where α represents the loudness attenuation coefficient and $0 < \alpha < 1$. σ represents the pulse frequency enhancement coefficient and $\sigma > 0$.

3.2. Cuckoo Search Algorithm. The cuckoo search algorithm (CS) is simplification and simulation of the cuckoo nest finding and spawning behavior. The special habit of cuckoos is parasitic brooding; that is, other host birds hatch and brood on their behalf. In order to make this phenomenon difficult to detect, the bird will first find a bird with similar characteristics to its own egg as the host during the breeding period. After being recognized by the host bird, the egg is removed or the host rebuilds the nest. In order to simulate its reproductive behavior, the following assumptions are proposed in the process:

- (1) Each cuckoo lays only one egg at a time and randomly selects the nest to hatch
- (2) The best bird's nest is retained to the next generation
- (3) The number of available bird nests n remains unchanged; there is a probability (p_a) that the host bird finds foreign eggs, $p_a \in [0, 1]$

For the cuckoo search algorithm, randomly initialize n bird nest positions in the d -dimensional search space and leave the best position to the next generation. The new position is generated by Levy flight. Then the cuckoo's nest search path and position update formula are as follows:

$$x_i^{t+1} = x_i^t + \alpha \oplus \text{Levy}(\lambda), \quad (15)$$

where x_i^t represents the position of the i -th bird nest in the t -th generation, α represents the step-length control factor

and $\alpha > 0$, \oplus is the point-to-point multiplication, $\text{Levy}(\lambda)$ is the random search path, and $\text{Levy} \sim u = t^{-\lambda}$ ($1 < \lambda \leq 3$).

After the position is updated, compare the random numbers r and p_a , and $0 \leq r \leq 1$; if $r > p_a$, then use the random walk method to change the position so as to retain a set of better values and obtain the current optimal bird nest position and optimal solution through iteration. The update formula is as follows:

$$x_i^{t+1} = x_i^t + \tau(x_m^t - x_k^t), \quad (16)$$

where τ represents the uniformly distributed scaling factor within $[0, 1]$ and both x_m^t and x_k^t represent the random solution in the t -th generation.

3.3. Bat Cuckoo Hybrid Algorithm. Although the bat algorithm has low convergence accuracy, its global search ability is strong; in order to improve the quality of the cuckoo population, the bat algorithm is integrated into the cuckoo algorithm for optimization, and a bat cuckoo hybrid algorithm (BACS) is proposed. For this algorithm, the nest position obtained by the cuckoo algorithm is not directly used as the initial position, but the bat algorithm is used to continue to optimize the optimal value after the position is updated, which greatly accelerates the global search ability of the algorithm. Therefore, the integration of the two algorithms effectively balances the problem of local and global optimization. Based on this, the specific steps of the bat cuckoo hybrid algorithm are shown in Table 1.

4. Hybrid Algorithm of Extreme Learning Machine Based on Bat Cuckoo Algorithm

Extreme Learning Machine (ELM) selects hidden layer parameters randomly and does not need to update iteratively during training, and the output weight can be determined by the least square solution, which greatly accelerates the learning process. Although ELM overcomes the shortcomings of the traditional gradient descent algorithm, the number of hidden nodes still needs to be set in advance, which may lead to many redundant nodes. Therefore, ELM requires more random hidden nodes in some applications than traditional neural network algorithms. However, this will lead to a decrease in the sparsity and regulation ability of the hidden layer, the complexity of the network structure, and the extension of the training time and finally affect the generalization ability and robustness of the network.

TABLE 1: Steps of the bat cuckoo hybrid algorithm.

Step 1: initialize the basic parameters and set the loop termination criteria
Step 2: initialize the location of the bird nest, calculate the fitness value of each bird nest, and obtain the optimal position and optimal value
Step 3: record the optimal position of the previous generation, update according to formula (15) to obtain a new set of positions, calculate the fitness value, and compare it with the value of the previous generation to determine the current better position
Step 4: compare the random number r with p_a ; if $r > p_a$, update the position randomly; otherwise, it will not change
Step 5: use the new position as the initial point of the bat algorithm and use equations (9)–(14) to update the position of the bird nest
Step 6: record the position of step 5 and calculate the fitness value to determine the current optimal position and optimal value
Step 7: if the termination conditions are met, continue to the next step; otherwise, go to step 3
Step 8: output the global optimal position, and the algorithm ends

BACS algorithm has the characteristics of strong search accuracy, fast convergence speed, and not easy to fall into local best and effectively balances local and global search. Using this optimization ability, the hidden layer parameters of ELM are selected appropriately to solve the problem that the hidden layer parameters need to be optimized due to randomness. Therefore, this paper considers the use of the BACS algorithm to optimize ELM so as to propose a hybrid algorithm of Extreme Learning Machine based on the bat cuckoo algorithm (BACS – ELM). We first use the BACS algorithm to train the input weights and thresholds randomly generated by ELM. The population is taken as the initially hidden layer parameter of ELM, and the fitness function of the BACS algorithm is used to conduct iterative optimization. The position of the individual of the population is constantly adjusted to find the optimal hidden layer parameter until the maximum number of iterations or search accuracy is reached. At the end of the iteration, the optimal individual position is obtained, and the optimized results are used as the input weights and thresholds of ELM to train the network so as to improve the convergence speed and stability of the network model. To prevent the problem of output saturation caused by excessive input value, we use the following formula to normalize the data:

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \quad (17)$$

where x is the original data and x_{\max} and x_{\min} represent the maximum and minimum values of the original data, respectively.

Next, the input weights and thresholds of ELM were represented by the cuckoo individuals using real coding rules. On the basis of Section 2, the number of neurons in the input layer and hidden layer is fixed as n and L , respectively. Therefore, the calculation formula of the coding length of the cuckoo individual is

$$D = (n + 1) * L. \quad (18)$$

Individual position of cuckoo can be expressed as

$$X = (x_1, x_2, x_3, \dots, x_{L \times n + L}). \quad (19)$$

The input weights w_i and thresholds b_i of ELM are mapped to the individual position of the cuckoo, the population is randomly initialized, and the obtained random individuals are assigned to the input weights and thresholds of ELM one by one and placed in the ELM network. Here, the assignments of input weights and thresholds are, respectively, expressed as follows:

$$\begin{aligned} w_i &= (x_1, x_2, x_3, \dots, x_{L \times n}), \\ b_i &= (x_{L \times n + 1}, x_{L \times n + 2}, \dots, x_{L \times n + L}). \end{aligned} \quad (20)$$

In the training sample process of ELM, in order to evaluate the prediction performance more objectively, we used the root mean square error as the evaluation index of model prediction, so the fitness function was designed as

$$\text{RMSE} = \sqrt{\frac{1}{P} \sum_{j=1}^P (t_j - o_j)^2}, \quad (21)$$

where P is the total number of samples, $T = (t_1, t_2, \dots, t_p)$ represents the actual output value of samples, and $O = (o_1, o_2, \dots, o_p)$ represents the expected output value of samples. Table 2 shows the specific implementation steps of the BACS – ELM algorithm.

5. Experimental Results

In order to verify the performance of the proposed algorithm, a function fitting and several classification problems are tested in this section, and the validity of BACS – ELM is tested by comparing it with the *ELM*, *BA – ELM*, and *CS – ELM* algorithms.

5.1. Function Fitting. In order to declare the performance of the proposed algorithm more intuitively and effectively, we take into account adopting ELM, BA – ELM, CS – ELM, and BACS – ELM to approximate the Sinc function and then compare the function approximation capabilities. The expression for the Sinc function is defined as follows:

$$f(x) = \begin{cases} \frac{\sin x}{x}, & x \neq 0, \\ 1, & x = 0. \end{cases} \quad (22)$$

The training set and test set of 5000 samples were selected, respectively, and the input variables x_i obey the uniform distribution in the interval $[-10, 10]$. In order to increase the authenticity and improve the generalization performance of the algorithm, random noise was added to the training samples, whereas the testing data remained noise-free. For different optimization methods, the initial parameter settings are presented in Table 3, and the maximum iteration number is set $I = 100$. The activation function is the RBF function, and the fitness function is

TABLE 2: Steps of the BACS – ELM learning algorithm.

Step 1: initialize the basic parameters and set the loop termination criteria
Step 2: initialize the cuckoo individual, code the input weights and thresholds of ELM into the individual, and each individual represents an ELM network structure
Step 3: normalize the training data and random initial individual position and calculate the fitness value in line with equation (21)
Step 4: record the optimal position, obtain a group of new positions according to equation (15), calculate the fitness value, and determine the current optimal position
Step 5: compare the random numbers r with P_a ; if $r > P_a$, update the position randomly; otherwise, it will not change
Step 6: take the new position as the starting point of BA, and randomly generate rand_1 ; if $\text{rand}_1 > r_i$, update the current optimal position; otherwise, go to step 7
Step 7: randomly generate rand_1 ; if $\text{rand}_1 < A_i$ and $f(x_{\text{new}}) > f(x_{\text{old}})$, replace x_{new} with the current position x_i^{t+1} or do not update x_{new}
Step 8: calculate the fitness value of each individual, and determine the current optimal position and optimal value
Step 9: if the termination condition is met, proceed to the next step; otherwise, go to step 4
Step 10: the individual cuckoo is decoded into the input weights and thresholds of ELM; obtain the optimal ELM network structure according to these parameters

TABLE 3: The population parameter setting of three optimization methods.

Optimization algorithm	Parameter setting
BA – ELM	Bat population size $N = 20$, loudness $A_0 = 1$, and pulse emissivity $r = 0.5$
CS – ELM	Loudness attenuation coefficient $\alpha = 0.9$ and pulse frequency interval $[f_{\min}, f(\max)] = [0, 2]$ Initial nest size $N = 20$, discovery probability $P_a = 0.25$, and step size control factor $\alpha = 0.5$ Scaling factor $\gamma = 0.5$
BACS – ELM	Bat population size $N = 20$, loudness $A_0 = 1$, and pulse emissivity $r = 0.5$ Loudness attenuation coefficient $\alpha = 0.9$ and pulse frequency interval $[f_{\min}, f(\max)] = [0, 2]$ Discovery probability $P_a = 0.25$, step size control factor $\alpha = 0.5$, and scaling factor $\gamma = 0.5$

RMSE. In order to compare the results of each algorithm more objectively, each experiment was run 20 times and then took the mean value.

The selection of the number of hidden nodes will have a direct influence on the performance of the model. Therefore, the experiment on BACS – ELM was carried out by adjusting the number of hidden nodes, and the test results obtained are shown in Table 4. The results show that the function has the best fitting effect when the number of hidden nodes is 12, and the mean square error of training and testing tends to be stable with the increase of nodes. To ensure the performance of the algorithm and reduce the complexity of the model, the architecture of the optimized ELM network can be determined as 1-12-1.

Then, based on the selection of the above parameter values, simulation experiments were carried out on the ELM, BA – ELM, CS – ELM, and BACS – ELM algorithms. It can be seen from Figure 2 that the approximation effect of the BACS – ELM algorithm is better than that of other algorithms. Moreover, the performance comparison of each algorithm is shown in Table 5. According to the displayed results, the test RMSE value of the BACS – ELM algorithm is the smallest, which means that the algorithm has higher accuracy and better stability. As can be seen from the training time in the table, due to the randomness of hidden layer parameters of ELM, it has a very fast learning speed, but the fitting effect is not ideal.

The results also show that the three optimization methods are all effective. But there is little difference in training and testing time between the BA – ELM, CS – ELM, and BACS – ELM algorithms and the advantages of learning efficiency are not embodied. Nevertheless, the ELM model

based on the BACS algorithm greatly improves the convergence accuracy of function fitting, so the computational efficiency is also within the acceptable range.

5.2. Classification Problems. In this section, in order to more accurately appraise the effectiveness of the BACS – ELM algorithm, the performance of the algorithm will be compared on multiple classification problems. The relevant information of the dataset is given in Table 6. The initial parameter setting of each group was consistent with the above. The maximum iterations number $I = 100$ and the activation function was the Sigmoid function. Each group of experiments was run 20 times to take the average value.

Figure 3 shows the comparison of the classification accuracy of the algorithm in different datasets with the change of the number of nodes. Figure 3(a) is based on the variation trend of breast cancer; it can be seen from the figure that ELM needs the most nodes to achieve relatively high accuracy, while other algorithms all achieve the highest accuracy when the node is 20, and further speaking, BACS – ELM is slightly better. Figure 3(b) is based on the changing trend of heart failure. It can be seen from the figure that the four algorithms all show a similar curve trend when the number of hidden nodes increases and they all have the best accuracy when the node is 20, but at this time, BACS – ELM has the highest value of 84.23%. Figure 3(c) is based on the variation trend of Iris. BACS – ELM has the best accuracy when the node is 10, which is 5 fewer nodes than other algorithms when they get the maximum value. Figure 3(d) is based on the changing trend of the vertebral column. It can be seen from the graph that BACS – ELM only needs the

TABLE 4: The influence of hidden node number on BACS – ELM algorithm.

Number of hidden nodes	2	4	6	8	10	12	14	16
Training RMSE	0.1370	0.1183	0.1109	0.1094	0.1082	0.1081	0.1087	0.1081
Testing RMSE	0.0902	0.0516	0.0279	0.0098	0.0095	0.0084	0.0087	0.0085

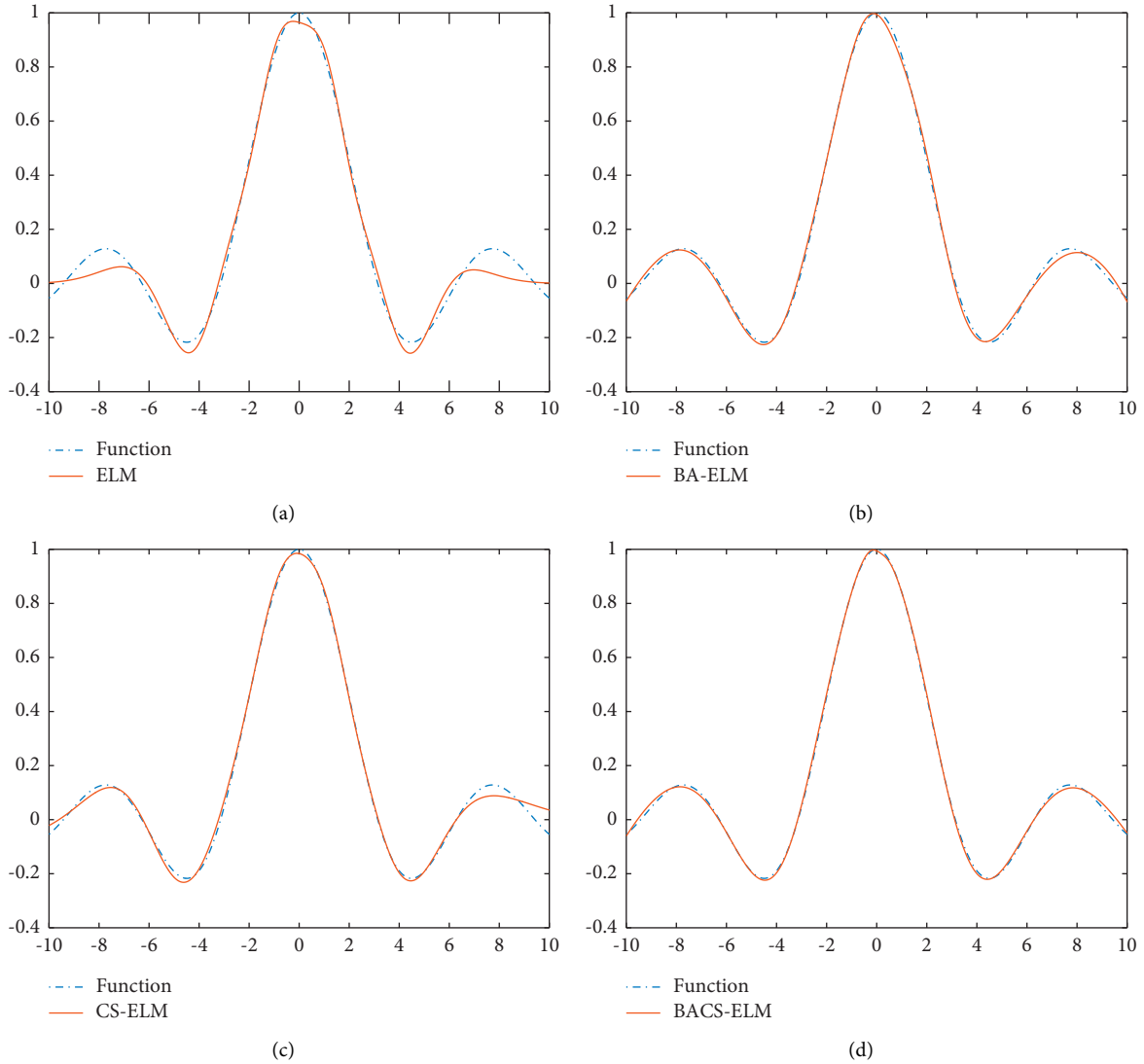


FIGURE 2: Comparison of the fitting effects of the four algorithms: (a) ELM, (b) BA – ELM, (c) CS – ELM, and (d) BACS – ELM.

TABLE 5: The performance comparison of four algorithms on Sinc function.

Algorithms	Training time (s)	Testing time (s)	Training RMSE	Testing RMSE
ELM	0.0056	0.0018	0.1351	0.0453
BA – ELM	40.6553	0.0019	0.1095	0.0119
CS – ELM	35.9939	0.0022	0.1107	0.0148
BASC – ELM	42.7807	0.0026	0.1081	0.0084

minimum number of nodes to obtain the best results, and the accuracy value fluctuates little, which indicates that the algorithm can achieve better stability.

Next, in order to better explain the accuracy of the BACS – ELM algorithm in classification experiments,

Figure 4 presents the fitness curves of the BA – ELM, CS – ELM, and BACS – ELM algorithms under four classification problems, respectively. To maintain the consistency of the experimental environment, the number of hidden nodes for each problem was set as 20, 20, 15, and 30,

TABLE 6: The detailed description of the classification dataset.

Dataset	Training samples	Testing samples	Attribute	Classes
Breast cancer	80	36	9	2
Heart failure	209	90	12	2
Iris	105	45	4	3
Vertebral column	208	102	6	3

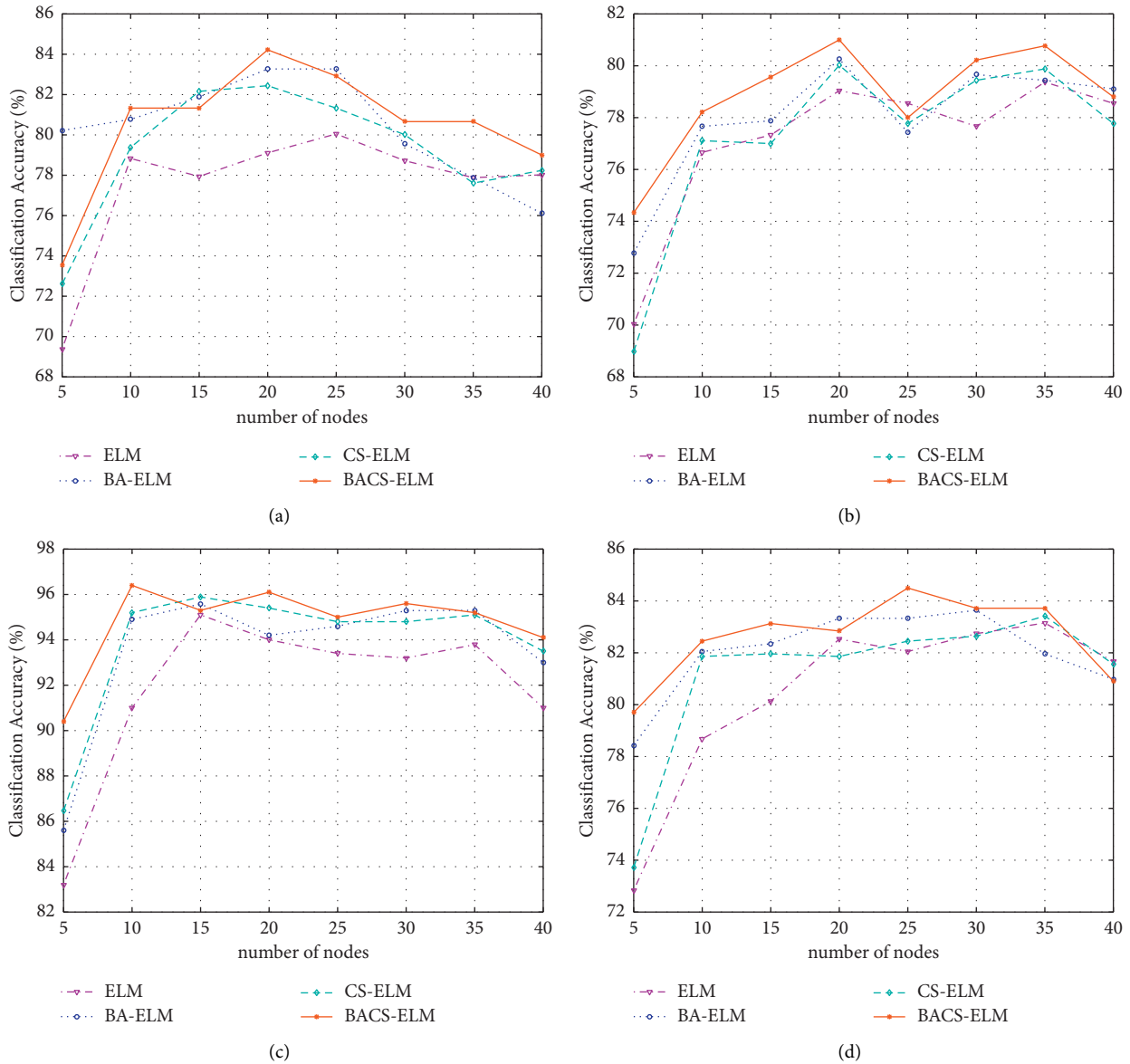


FIGURE 3: The graph of the classification accuracy over different datasets with the change of the number of nodes: (a) breast cancer; (b) heart failure; (c) iris; (d) vertebral column.

respectively, while other parameters were unchanged. As can be seen from Figures 4(a)–4(d), for different datasets, compared with the BA – ELM and CS – ELM algorithms, the BACS – ELM algorithm can obtain the best fitness function value in the case of the least number of iterations. This is because when the BACS algorithm optimizes the input weights and thresholds of ELM, it has a strong local

optimization ability at the initial stage of search and makes full use of the global optimization ability of the BA algorithm. The combination of the two greatly improves the convergence accuracy.

Based on the above analysis, the performance results of the four algorithms on the number of hidden nodes, training time, training, and test accuracy are also given in the

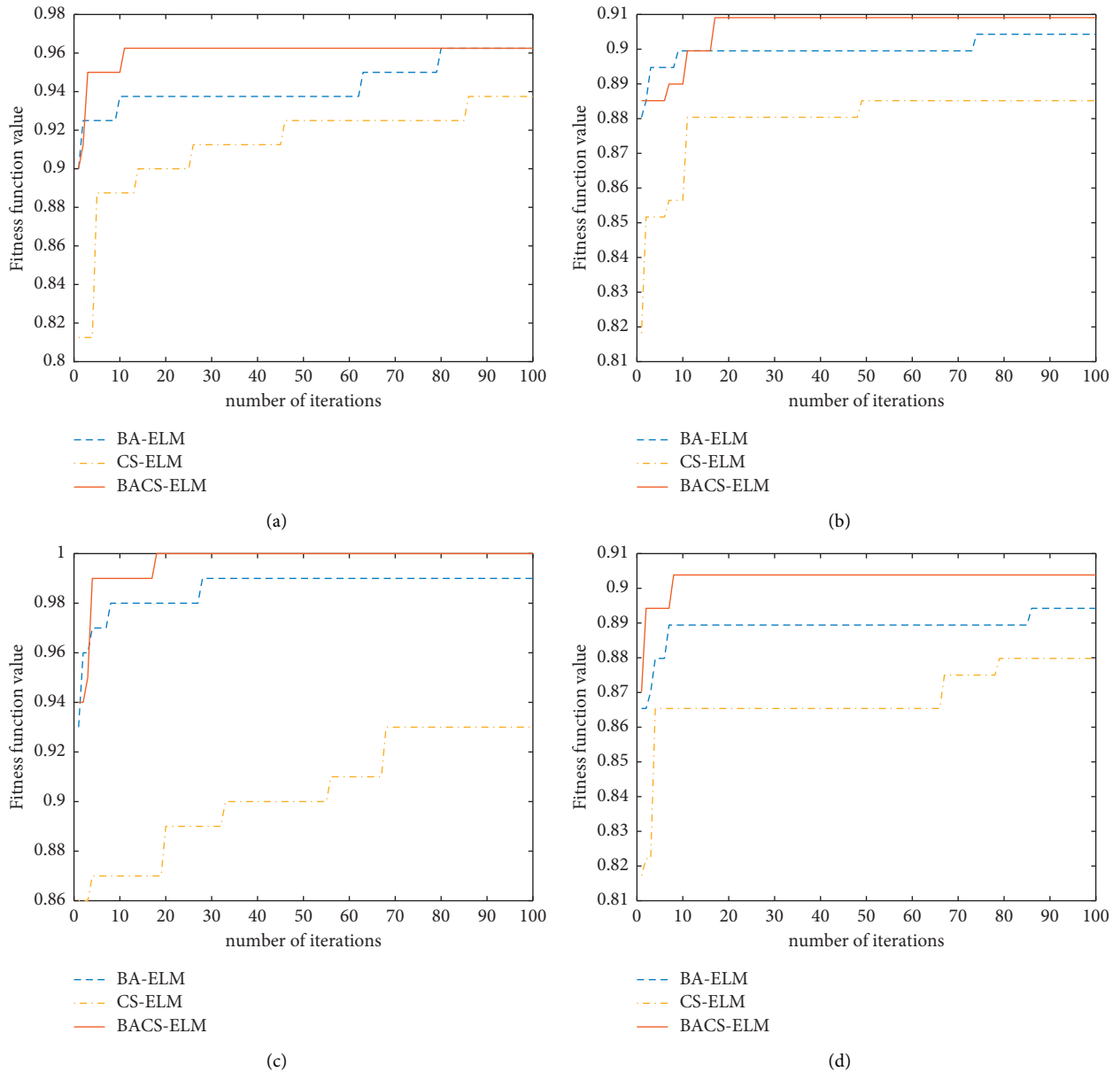


FIGURE 4: Comparison of fitness function curves on different datasets: (a) breast cancer; (b) heart failure; (c) iris; (d) vertebral column.

experiment. It can be clearly seen from Table 7 that the BACS – ELM algorithm can achieve the best test accuracy under the minimum number of hidden nodes in all the four datasets, which indicates that the algorithm can effectively optimize the parameters of the hidden layer of the ELM model by using BACS algorithm and then obtain a more appropriate and simplified network structure. At the same time, the best generalization performance and classification ability are obtained. In terms of computing time or efficiency, hidden layer parameters of ELM do not need to be

iteratively tuned, so the learning speed is very fast, but the success rate of its classification is very low. In Table 7, we did not list the test time data because the values of the four algorithms for different datasets in the experimental results are very low, and the size is similar; that is to say, the impact of the data on the overall experiment results cannot be regarded as an evaluation item. Compared with the other two optimization methods, although the BACS – ELM algorithm is slightly worse in learning efficiency, it shows great advantages in classification accuracy.

TABLE 7: Performance comparison for classification problems.

Datasets	Algorithm	Training accuracy (%)	Testing accuracy (%)	Number of hidden nodes	Train time (s)
Breast cancer	ELML	84.05	80.05	25	0.0046
	BA – ELM	87.65	83.27	20	7.8443
	CS – ELM	86.00	82.44	20	7.2638
	BACS – ELM	89.27	84.23	20	9.2870
Heart failure	ELML	82.26	79.04	20	0.0069
	BA – ELM	85.64	80.25	20	9.0520
	CS – ELM	84.35	80.02	20	8.1702
	BACS – ELM	86.29	81.00	20	10.236
Iris	ELML	96.79	95.10	15	0.0062
	BA – ELM	97.90	95.58	15	5.2547
	CS – ELM	97.00	95.89	15	5.8250
	BACS – ELM	98.40	96.40	10	6.7234
Vertebral column	ELML	85.33	82.74	35	0.0078
	BA – ELM	86.87	83.64	30	10.568
	CS – ELM	86.82	83.42	35	11.534
	BACS – ELM	88.07	84.50	25	13.939

6. Conclusions

In this paper, we propose a hybrid Extreme Learning Machine algorithm based on the bat and cuckoo search algorithm to optimize the input weight and threshold of the traditional ELM algorithm, thus improving the disadvantages of traditional ELM, such as poor sparsity of hidden layer, low adjustment ability, and complex network structure. Meanwhile, the BACS algorithm has the characteristics of strong searching accuracy, fast convergence speed, and not easy to fall into the local optimal, which effectively balances the local and global optimization problems. Therefore, the proposed BACS-ELM algorithm can effectively solve the optimization problem due to the randomness of hidden layer parameters and improve the generalization performance of the network.

Experimental results show that the BACS-ELM algorithm is superior to other algorithms in function fitting and classification. In the future, we consider extending the BACS-ELM algorithm to practical application problems and solving a wider class of even tougher optimization problems.

Data Availability

All data included in this study are available upon request by contact with the corresponding author.

Disclosure

This manuscript is the authors' original work and has not been published nor has it been submitted simultaneously elsewhere.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was funded by the Special Science Research Plan of the Education Bureau of Shaanxi Province of China (no.

18JK0344) and the Natural Science Basic Research Plan in Shaanxi Province of China (no. 2021JM-446).

References

- [1] D. Liu, W. Baskett, D. Q. Beversdorf, and C. R. Shyu, "Exploratory data mining for subgroup cohort discoveries and prioritization," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, pp. 1456–1468, 2019.
- [2] I. Yasin, V. Drga, F. Liu, A. Demosthenous, and R. Meddis, "Optimizing speech recognition using a computational model of human hearing: effect of noise type and efferent time constants," *IEEE Access*, vol. 8, pp. 56711–56719, 2020.
- [3] N. T. An, P. D. Dong, and X. Qin, "Robust feature selection via nonconvex sparsity-based methods," *Journal of Nonlinear and Variational Analysis*, vol. 5, pp. 59–77, 2021.
- [4] Q. Kang, Q. Fan, and J. M. Zurada, "Deterministic convergence analysis via smoothing group lasso regularization and adaptive momentum for sigma-pi-sigma neural network," *Information Sciences*, vol. 553, pp. 66–82, 2021.
- [5] S. S. Ho, M. Schofield, and N. Wang, "Learning incentivization strategy for resource rebalancing in shared services with a budget constraint," *Journal of Applied and Numerical Optimization*, vol. 3, pp. 105–114, 2021.
- [6] Y. Tsuruoka, "Deep learning and natural language processing," *Brain and Nerve*, vol. 71, pp. 45–55, 2019.
- [7] Q. Fan, J. Peng, H. Li, and S. Lin, "Convergence of a gradient-based learning algorithm with penalty for ridge polynomial neural networks," *IEEE Access*, vol. 9, pp. 28742–28752, 2021.
- [8] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 985–990, Budapest, Hungary, July 2004.
- [9] G. B. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 274–281, 2003.
- [10] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.
- [11] 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.

- [12] J. Ding, G. Chen, and K. Yuan, "Short-term wind power prediction based on improved grey wolf optimization algorithm for extreme learning machine," *Processes*, vol. 8, no. 1, p. 109, 2020.
- [13] N. Nabipour, A. Mosavi, A. Baghban, S. Shamshirband, and I. Felde, "Extreme learning machine-based model for solubility estimation of hydrocarbon gases in electrolyte solutions," *Processes*, vol. 8, no. 1, p. 92, 2020.
- [14] S. Saraswathi, S. Sundaram, N. Sundararajan, M. Zimmermann, and M. Nilsen-Hamilton, "ICGA-PSO-ELM approach for accurate multiclass cancer classification resulting in reduced gene sets in which genes encoding secreted proteins are highly represented," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 2, pp. 452–463, 2011.
- [15] W. Huang, Y. Yang, Z. Lin et al., "Random feature subspace ensemble based extreme learning machine for liver tumor detection and segmentation," in *Proceedings of the 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4675–4678, Chicago, IL, USA, August 2014.
- [16] 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, pp. 302–310, 2018.
- [17] Z. Chen, L. Wu, S. Cheng, P. Lin, Y. Wu, and W. Lin, "Intelligent fault diagnosis of photovoltaic arrays based on optimized kernel extreme learning machine and I–V characteristics," *Applied Energy*, vol. 204, pp. 912–931, 2017.
- [18] G. Xu-Sheng, Q. Hong, M. Xiang-Wei, W. Chun-Lan, and Z. Jie, "Research on ELM soft fault diagnosis of analog circuit based on KSLPP feature extraction," *IEEE Access*, vol. 7, pp. 92517–92527, 2019.
- [19] H. Zou, B. Huang, X. Lu, H. Jiang, and L. Xie, "A robust indoor positioning system based on the procrustes analysis and weighted extreme learning machine," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1252–1266, 2016.
- [20] M. Zhang, Y. Wen, J. Chen, X. Yang, R. Gao, and H. Zhao, "Pedestrian dead-reckoning indoor localization based on OS-ELM," *IEEE Access*, vol. 6, pp. 6116–6129, 2018.
- [21] X. Tang and M. Han, "Partial lanczos extreme learning machine for single-output regression problems," *Neurocomputing*, vol. 72, no. 13–15, pp. 3066–3076, 2009.
- [22] Q. W. Fan and T. Liu, "Smoothing L_0 regularization for extreme learning machine," *Mathematical Problems in Engineering*, vol. 2020, Article ID 9175106, 10 pages, 2020.
- [23] Q. Fan, L. Niu, and Q. Kang, "Regression and multiclass classification using sparse extreme learning machine via smoothing group $L_{1/2}$ regularizer," *IEEE Access*, vol. 8, pp. 191482–191494, 2020.
- [24] Q.-Y. Zhu, A. K. Qin, P. N. Suganthan, and G.-B. Huang, "Evolutionary extreme learning machine," *Pattern Recognition*, vol. 38, no. 10, pp. 1759–1763, 2005.
- [25] S. Salcedo-Sanz, A. Pastor-Sánchez, L. Prieto, A. Blanco-Aguilera, and R. García-Herrera, "Feature selection in wind speed prediction systems based on a hybrid coral reefs optimization—extreme learning machine approach," *Energy Conversion and Management*, vol. 87, pp. 10–18, 2014.
- [26] Q. L. Ling and F. Han, "Improving the conditioning of extreme learning machine by using particle swarm optimization," *International Journal of Computer Technology and Applications*, vol. 6, pp. 85–93, 2012.
- [27] Y. Xu and Y. Shu, "Evolutionary extreme learning machine—based on particle swarm optimization," *Advances in Neural Networks*, vol. 3971, pp. 644–652, 2006.
- [28] S. Suresh, R. Venkatesh Babu, and H. J. Kim, "No-reference image quality assessment using modified extreme learning machine classifier," *Applied Soft Computing*, Elsevier, vol. 9, pp. 541–552, Amsterdam, Netherlands, 2009.
- [29] X. S. Yang, *Nature-Inspired Meta-Heuristic Algorithms*, Luniver Press, Stansted Mountfitchet, UK, 2010.
- [30] X. S. Yang and S. Deb, "Cuckoo search via Levy flights," in *Proceedings of the IEEE World Congress on Nature and Biologically Inspired Computing*, pp. 210–214, Coimbatore, India, 2009.
- [31] X. S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, no. 4, pp. 330–343, 2010.
- [32] R. Rajabioun, "Cuckoo optimization algorithm," *Applied Soft Computing*, vol. 11, no. 8, pp. 5508–5518, 2011.
- [33] P. Civicioglu and E. Besdok, "A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms," *Artificial Intelligence Review*, vol. 39, no. 4, pp. 315–346, 2013.
- [34] M. K. Marichelvam, "An improved hybrid cuckoo search (IHCS) metaheuristics algorithm for permutation flow shop scheduling problems," *International Journal of Bio-Inspired Computation*, vol. 4, no. 4, pp. 200–205, 2012.
- [35] E. Valian, S. Mohanna, and S. Tavakoli, "Improved cuckoo search algorithm for feed forward neural network training," *International Journal of Artificial Intelligence & Applications*, vol. 2, no. 3, pp. 36–43, 2011.
- [36] X. S. Yang and A. H. Gandomi, *Bat Algorithm: A Novel Approach for Global Engineering Optimization*, Professional Publications, Hyderabad, India, 2012.
- [37] A. H. Gandomi and X.-S. Yang, "Chaotic bat algorithm," *Journal of Computational Science*, vol. 5, no. 2, pp. 224–232, 2014.
- [38] X. S. Yang, "Cuckoo search and firefly algorithm: theory and applications," *Studies in Computational Intelligence*, p. 516, Springer, Heidelberg, Germany, 2013.
- [39] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.
- [40] A. Gálvez and A. Iglesias, "Memetic improved cuckoo search algorithm for automatic B-spline border approximation of cutaneous melanoma from macroscopic medical images," *Advanced Engineering Informatics*, vol. 43, 2020.
- [41] X.-S. Yang and S. Deb, "Multiobjective cuckoo search for design optimization," *Computers & Operations Research*, vol. 40, no. 6, pp. 1616–1624, 2013.
- [42] A. M. Kamoona and J. C. Patra, "A novel enhanced cuckoo search algorithm for contrast enhancement of gray scale images," *Applied Soft Computing*, vol. 85, Article ID 105749, 2019.
- [43] K. Khan and A. Sahai, "A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context," *International Journal of Intelligent Systems and Applications*, vol. 4, no. 7, pp. 23–29, 2012.
- [44] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.