Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 7273061, 13 pages
https://doi.org/10.1155/2017/7273061

Research Article

Radar Target Classification Using an Evolutionary Extreme Learning Machine Based on Improved Quantum-Behaved Particle Swarm Optimization

Feixiang Zhao, Yongxiang Liu, Kai Huo, and Zhongshuai Zhang

College of Electronic Science, National University of Defense Technology, No. 109 Deya Road, Kaifu District, Changsha 410073, China

Correspondence should be addressed to Yongxiang Liu; lyx_bible@sina.com

Received 20 June 2017; Accepted 2 November 2017; Published 6 December 2017

Academic Editor: Luciano Mescia

A novel evolutionary extreme learning machine (ELM) based on improved quantum-behaved particle swarm optimization (IQPSO) for radar target classification is presented in this paper. Quantum-behaved particle swarm optimization (QPSO) has been used in ELM to solve the problem that ELM needs more hidden nodes than conventional tuning-based learning algorithms due to the random set of input weights and hidden biases. But the method for calculating the characteristic length of Delta potential well of QPSO may reduce the global search ability of the algorithm. To solve this issue, a new method to calculate the characteristic length of Delta potential well is proposed in this paper. Experimental results based on the benchmark functions validate the better performance of IQPSO against QPSO in most cases. The novel algorithm is also evaluated by using real-world datasets and radar data; the experimental results indicate that the proposed algorithm is more effective than BP, SVM, ELM, QPSO-ELM, and so on, in terms of real-time performance and accuracy.

1. Introduction

Radar target classification technology is of great significance in both military and civil aspects [1, 2]. At present, the commonly used classification methods include Bayesian [3], Dempster-Shafer (D-S) theory [4], decision tree [5], support vector machine (SVM) [6], and back propagation (BP) neural network [7]. Although these methods can obtain good classification accuracy, their common problem is that the real-time performance is not high. ELM [8], as a new learning algorithm for single-hidden-layer feedforward neural networks (SLFNs), has attracted great concerns from various fields for its fast learning speed, such as traffic sign recognition [9], face recognition [10], human action recognition [11], and image analysis [12]. The idea of the algorithm is to generate the input weights and hidden biases randomly and then train the network by solving the norm least-squares solution of the output weights [13]. ELM not only has faster learning speed than traditional learning methods, but also has good generalization performance in many applications [14]. In order to further improve the ELM algorithm, the researchers put forward many improved algorithms. The fully complex ELM (C-ELM) is proposed in [15], which extends the ELM algorithm from the real domain to the complex domain. Considering that when new data is received, many training methods use the past data together with the new data and perform a retraining, which consumes lots of time, an online sequential ELM (OS-ELM) is proposed in [16], which can learn the training data one by one or chunk by chunk and discard the data for which the training has already been done. To get the better predicting performance, a new adaptive ensemble model of ELM (Ada-ELM) is proposed in [17], which can adjust the ensemble weights automatically. Considering that the performance of ELM is affected by hidden layer nodes and the number of hidden layer nodes is difficult to determine, the incremental ELM (I_ELM) [18], pruned ELM (P_ELM) [19], and self-adaptive ELM (SaELM) [20] have been proposed. Note that traditional ELM only utilizes labeled data to carry out the supervised learning task; [21] applied manifold regularization (MR) to ELM to exploit unlabeled data in the ELM model. Moreover, [22] proposed sparse Bayesian ELM (SBELM) which has the advantages of the two algorithms.

However, due to the random selection of input weights and hidden biases, ELM tends to need a large number of hidden nodes for better generalization, which may increase the complexity of the network. In order to solve this problem, an improved ELM method based on particle swarm optimization (PSO) is proposed in [23]. This method can resolve the drawbacks of ELM. But, at the same time, we also know that PSO easily has premature convergence and has low robustness due to the fact that its global search ability relies on the up-limit of velocity [24]. To discourage premature convergence, [25] proposed comprehensive learning particle swarm optimizer (CLPSO), which utilized all other particles' historical best information to update a particle's velocity. In [26], Sun et al. introduce quantum theory into PSO and put forward a quantum-behaved PSO (QPSO) algorithm, which outperforms PSO in search ability and has fewer parameters to control. QPSO and its improved model have been applied in many ways [27, 28]. The enhanced weighted quantum PSO (EWQPSO) has been developed to perform the design of the supershaped lens antennas yielding optimal antenna performance [27] and the random local optimized QPSO (RLQPSO) has been used in fast threshold image segmentation [28]. In [29–31], QPSO was applied to ELM to improve the algorithm performance. However, as with other evolutionary algorithms, QPSO also has the problem of premature convergence [32]. References [33, 34] proposed QPSO with random mean best position and weighted mean best position, respectively. Reference [35] introduced a novel search strategy with a selection operation into QPSO. In the modified QPSO (MQPSO), the global best position is substituted by a personal best position of a randomly selected particle. Although QPSO algorithm has better global convergence than PSO, this method calculates the characteristic length of Delta potential well only according to the mean best position, which will reduce the global search ability of the algorithm. In order to overcome this problem, this paper presents a new method to calculate the characteristic length of Delta potential well. Then the improved QPSO (IQPSO) algorithm is used to optimize the weights and biases of ELM.

The rest of this paper is organized as follows: Section 2 introduces the relevant theoretical knowledge of ELM and QPSO. In Section 3, we present the improved formula for calculating the characteristic length of Delta potential well of QPSO and the application of IQPSO in parameter optimization of ELM. Experimental results are analyzed in Section 4, and Section 5 summarizes the paper.

2. Related Work

2.1. Extreme Learning Machine. Given a set of N training datasets (x_i, y_i), where $i = 1, 2, \ldots, N$, $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \in R^n$, and $y_i = [y_{i1}, y_{i2}, \ldots, y_{im}]^T \in R^m$, x_i is an n-dimensional input vector and y_i is the expected output. The output function of ELM with L hidden nodes is represented as follows:

$$\sum_{i=1}^{L} \beta_i g\left(w_i \cdot x_j + b_i\right) = o_j \quad j = 1, 2, \ldots, N, \tag{1}$$

where $w_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T \in R^n$ is the weight vector of input nodes to hidden nodes and b_i is the bias of ith hidden node, $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}] \in R^m$ is the weight vector between hidden nodes and the output nodes, $g(x)$ is the activation function of hidden layer, and o_j is the output vector.

If the SLFNs with L hidden nodes can approximate the N samples with zero error, we know that (1) can be converted to the following formula:

$$\begin{gathered} \sum_{i=1}^{L} \beta_i g\left(w_i \cdot x_1 + b_i\right) = y_1, \\ \vdots \\ \sum_{i=1}^{L} \beta_i g\left(w_i \cdot x_N + b_i\right) = y_N. \end{gathered} \tag{2}$$

The above equations can be written as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{Y}, \tag{3}$$

where

$$\begin{aligned} &\mathbf{H}\left(w_1, \ldots, w_L, b_1, \ldots, b_L, x_1, \ldots, x_N\right) \\ &\quad = \begin{bmatrix} g\left(w_1 \cdot x_1 + b_1\right) & \cdots & g\left(w_L \cdot x_1 + b_L\right) \\ \vdots & \cdots & \vdots \\ g\left(w_1 \cdot x_N + b_1\right) & \cdots & g\left(w_L \cdot x_N + b_L\right) \end{bmatrix}_{N \times L}, \end{aligned} \tag{4}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \qquad \mathbf{Y} = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N \times m}. \tag{5}$$

So training the SLFNs corresponds to finding the norm least-squares solution $\widehat{\boldsymbol{\beta}}$, which can be shown as follows:

$$\widehat{\boldsymbol{\beta}} = \mathbf{H}^{+}\mathbf{Y}, \tag{6}$$

where $\mathbf{H}^{+}$ is the Moore-Penrose generalized inverse of hidden layer output matrix $\mathbf{H}$.

Then, according to KKT theorem, (6) can be expressed as

$$\boldsymbol{\beta} = \begin{cases} \left(\dfrac{\mathbf{I}}{\lambda} + \mathbf{H}^{\mathbf{T}}\mathbf{H}\right)^{-1} \mathbf{H}^{T}\mathbf{Y}, & N > L, \\ \mathbf{H}^{T}\left(\dfrac{\mathbf{I}}{\lambda} + \mathbf{H}\mathbf{H}^{T}\right)^{-1} \mathbf{Y}, & N < L, \end{cases} \tag{7}$$

where $\mathbf{I}$ is the unit matrix and λ is the regularization coefficient.

Thus, the learning steps of the ELM can be summarized as follows.

Step 1. Determine the structure of neural networks and set random values to the input weights w_i and the hidden layer biases b_i.

Step 2. Calculate the hidden layer output matrix $\mathbf{H}$ according to (4).

Step 3. Calculate the output weight vector $\boldsymbol{\beta}$ according to (6).

2.2. Quantum-Behaved Particle Swarm Optimization. Given the particle swarm $S = \{s_1, s_2, \ldots, s_M\}$ where M represents the swarm size, D denotes the dimension of the search space. At each iteration, the position of the ith particle can be expressed as $s_i(t) = [s_{i1}(t), s_{i2}(t), \ldots, s_{iD}(t)]$. The personal best position can be expressed as $s_{pbest_i}(t) = [s_{pbest_{i1}}(t), s_{pbest_{i2}}(t), \ldots, s_{pbest_{iD}}(t)]$ $(i = 1, 2, \ldots, M)$ and the global best position can be expressed as $s_{gbest(t)} = [s_{gbest_1}(t), s_{gbest_2}(t), \ldots, s_{gbest_D}(t)]$. The personal best position and the global best position are updated using the following formula:

$$
\begin{gathered}
s_{pbest_i}(t+1) = \begin{cases} s_i(t) & f(s_i(t)) < f(s_{pbest_i}(t)), \\ s_{pbest_i}(t) & f(s_i(t)) \geq f(s_{pbest_i}(t)), \end{cases} \\
s_{gbest}(t+1) = \begin{cases} s_i(t) & f(s_i(t)) < f(s_{gbest}(t)), \\ s_{gbest}(t) & f(s_i(t)) \geq f(s_{gbest}(t)), \end{cases}
\end{gathered} \tag{8}
$$

where $f(x)$ is the fitness function.

In [30], it can be seen that the position of the particle can be updated by using (9)

$$s_{i,j}(t+1) = p_{i,j}(t) \pm \frac{L_{i,j}(t)}{2} \cdot \ln\left[\frac{1}{u_{i,j}(t)}\right], \tag{9}$$

$$
\begin{aligned}
p_{i,j}(t) = {} & \varphi_{i,j}(t) \cdot s_{pbest_{i,j}}(t) + (1 - \varphi_{i,j}(t)) \\
& \cdot s_{gbest_j}(t),
\end{aligned} \tag{10}
$$

where u and $\varphi_{i,j}(t)$ are two random numbers uniformly distributed in the interval (0,1) and $L_{i,j}(t)$ is the characteristic length of Delta potential well, and its value is directly related to the convergence speed and search ability of the algorithm. In the traditional QPSO, $L_{i,j}(t)$ is determined by

$$L_{i,j}(t) = 2\alpha \left| mbest_j(t) - s_{i,j}(t) \right|, \tag{11}$$

where $mbest$ is the mean best position of the swarm and $mbest_j(t) = \sum_{i=1}^{m} s_{pbest_{i,j}}(t)/M$. α is the contraction-expansion coefficient, which can be adjusted to control the convergence speed of the algorithm. There are two ways to set α: fixed parameter and linear reduction. If the linear reduction method is adopted, then $\alpha = 1 - 0.5 \times t/T_{\max}$, where $T_{\max}$ is the maximum number of iterations and t is the current number of iterations.

3. Proposed Approach

3.1. Improved Quantum-Behaved Particle Swarm Optimization. The method of (11) has the following problems: since $mbest$ is a relatively stable value, the search information of the whole swarm cannot be effectively utilized. Especially when most particles fall into local optima, only a few particles are distributed in other regions; the $mbest$ will also tend to the local optimum. As a result, most particles only perform local search. This will reduce the global search ability of the algorithm. To overcome this problem, this paper presents a new method to calculate $L_{i,j}(t)$.

$$
L_{i,j}(t) = \begin{cases}
\alpha \left| mbest_j(t) - \max(s_{i,j}(t), s_{k,j}(t)) \right|, & f_{\text{aver}} > \max(f(s_i(t)), f(s_k(t))), \\
\alpha \left| s_{i,j}(t) - s_{k,j}(t) \right|, & \min(f(s_i(t)), f(s_k(t))) < f_{\text{aver}} < \max(f(s_i(t)), f(s_k(t))), \\
\alpha \left| mbest_j(t) - \min(s_{i,j}(t), s_{k,j}(t)) \right|, & f_{\text{aver}} < \min(f(s_i(t)), f(s_k(t))),
\end{cases} \tag{12}
$$

where $s_{k,j}(t)$ is the randomly selected particle in the swarm and $k \neq i$, f_{aver} is the average fitness of the swarm, and $f_{\text{aver}} = \sum_{i=1}^{m} f(s(i))/M$.

Compared with (11), (12) takes into account the particles distributed in other regions. It can be seen from (12) that even if most particles fall into local optima, the particles still have moderate probability of jumping out of local optima due to the particles distributed in other regions. The improved method is able to improve the global search ability of QPSO.

The flow chart of IQPSO is shown in Figure 1.

3.2. ELM Based on IQPSO Algorithm. We know from Section 2 that ELM has the advantages of fast learning speed and easy implementation. However, ELM tends to require a large number of hidden nodes to get better performance, which complicates the network structure. In addition, the hidden layer output weights are determined by the random input weights and biases; as a result, the final output may be instable. This paper proposed IQPSO-ELM algorithm to solve the shortcoming of ELM.

The IQPSO-ELM can be divided into three parts: initial ELM, trained ELM, and test ELM as shown in Figure 2. Each part of the algorithm is summarized as follows.

(1) Initial ELM: we first obtain training samples, validation samples, and test samples. Then set the number of input nodes of ELM equal to the dimension of input data and set the number of output nodes equal to the number of sample classes. Of course, we also need to set hidden layer nodes, as well as the swarm size and the maximum number of iterations

(2) Trained ELM: in this process, appropriate input weights and hidden layer biases are obtained by IQPSO to train ELM. First, the particle dimension D

Table 1: Benchmark functions.

	Function expression	Range	Function optimal value
Sphere	$f_1(x) = \sum_{i=1}^{m} x_i^2$	$[-100, 100]$	0
Rosenbrock	$f_2(x) = \sum_{i=1}^{m-1} \left(100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2\right)$	$[-30, 30]$	0
Ackley	$f_3(x) = -20\exp\left(-0.2\sqrt{\left(\frac{1}{n}\sum_{i=1}^{n} x_i^2\right)}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right) + 20 + e$	$[-32, 32]$	0
Rastrigin	$f_4(x) = \sum_{i=1}^{n}\left[x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right]$	$[-5.12, 5.12]$	0

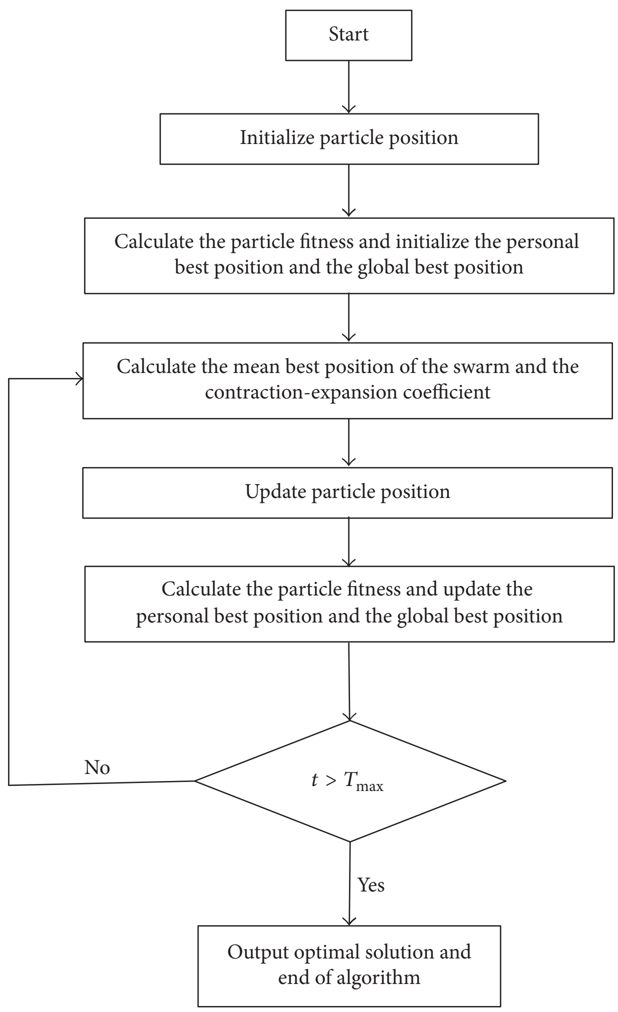

Figure 1: Flow chart of IQPSO algorithm.

is calculated according to the formula $D = D_{input} \times L + L$, where D_{input} and L represent the number of input nodes and hidden layer nodes, respectively. Then the position s_i of particle i ($i = 1, 2, \ldots, M$) is randomly initialized according to the particle dimension D and swarm size M. s_i can be written as $s_i = [w_{11}, \ldots, w_{LD_{input}}, b_1, b_2, \ldots, b_L]$. Finally, according to IQPSO introduced in Section 3.1, we can get the global best position s_{gbest}, which can be converted to the input weights and hidden layer biases (w_i, b_i) of ELM. What needs to be explained is that the fitness of IQPSO is represented by the correct classification rate.

(3) Test ELM: test samples are used to evaluate the effectiveness of the proposed method.

4. Experimental Result and Discussion

In this section, we will verify the effectiveness of the proposed algorithm. The experiments are performed on the Intel(R) Core(TM) 3.60 GHz CPU, with 8 GB of RAM, and Matlab R2013a environment.

4.1. The Performance of IQPSO on Benchmark Functions. In order to demonstrate the effectiveness of IQPSO, four benchmark functions (see Table 1) are selected for the experiment. The performance of IQPSO is compared with QPSO [26] and MQPSO [35]. We set different swarm size M for the four benchmark functions with different dimensions. M values are 20, 40, and 80. The maximum number of iterations is set as 1000, 1500, and 2000 corresponding to the dimensions 10, 20, and 30 for the four benchmark functions, respectively. The value of contraction-expansion coefficient α decreases from 1.0 to 0.5 linearly. The mean values of the best fitness values for 50 runs of each function are recorded in Tables 2–5, respectively.

According to Tables 2–5, the results show that the IQPSO works better than QPSO and MQPSO on Sphere function. IQPSO can also generate better results than the other methods in most cases on Rosenbrock function. On Ackley function, we know that the proposed method outperforms the QPSO but is not as good as MQPSO when the swarm size is 80. In addition, on Rastrigin function, the IQPSO is superior to QPSO and MQPSO except when the swarm size is 40 and dimension is 10. Generally speaking, the results show that the IQPSO has better global search ability than QPSO and MQPSO.

Figure 3 shows the convergence process of QPSO, MQPSO, and IQPSO on the four benchmark functions when the swarm size is 20, the dimension is 30, and the number of iterations is about 2000. It can be seen from the figures

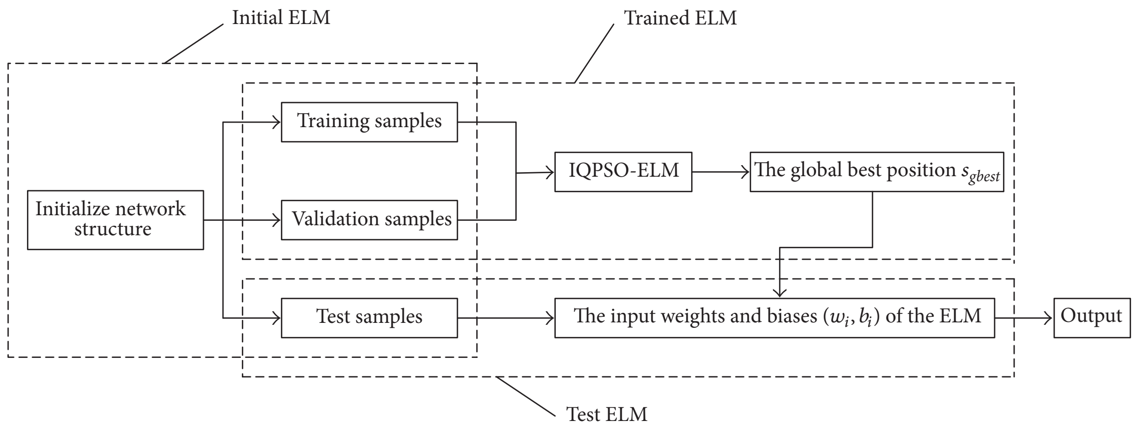

Figure 2: Flow chart of IQPSO-ELM algorithm.

Table 2: Test results of Sphere function.

M	Dimension	Iteration	QPSO	MQPSO	IQPSO
20	10	1000	7.7856e − 44	6.1122e − 39	**7.7399e − 53**
20	20	1500	2.0155e − 23	6.7054e − 20	**4.5967e − 44**
20	30	2000	1.3169e − 14	4.5319e − 13	**6.3732e − 44**
40	10	1000	9.4223e − 76	1.6406e − 62	**1.0500e − 83**
40	20	1500	1.4941e − 42	1.7135e − 38	**2.3700e − 62**
40	30	2000	4.4449e − 30	7.2975e − 28	**3.6540e − 59**
80	10	1000	4.1852e − 102	6.4074e − 75	**7.7866e − 128**
80	20	1500	1.7533e − 69	4.2385e − 55	**5.3777e − 89**
80	30	2000	9.5185e − 49	4.5975e − 42	**4.9417e − 78**

Table 3: Test results of Rosenbrock function.

M	Dimension	Iteration	QPSO	MQPSO	IQPSO
20	10	1000	6.9194	**5.8475**	9.4985
20	20	1500	39.7105	38.1779	**30.1639**
20	30	2000	55.3889	61.9106	**34.7486**
40	10	1000	**3.8067**	4.6050	5.2745
40	20	1500	31.3511	21.8120	**18.5787**
40	30	2000	44.2303	44.8217	**32.2876**
80	10	1000	4.1268	**3.3062**	3.8372
80	20	1500	30.9957	24.6872	**14.7379**
80	30	2000	37.2249	31.8952	**24.0227**

Table 4: Test results of Ackley function.

M	Dimension	Iteration	QPSO	MQPSO	IQPSO
20	10	1000	5.0093e − 15	**4.6541e − 15**	6.8567e − 15
20	20	1500	4.0184e − 12	3.1785e − 11	**1.5667e − 14**
20	30	2000	2.1237e − 8	5.7514e − 8	**2.3412e − 14**
40	10	1000	4.4409e − 15	4.3698e − 15	**4.0962e − 15**
40	20	1500	9.0594e − 15	8.5620e − 15	**7.9226e − 15**
40	30	2000	7.1374e − 14	1.0768e − 13	**1.6094e − 14**
80	10	1000	4.3698e − 15	**4.2988e − 15**	4.4409e − 15
80	20	1500	7.6383e − 15	**6.2883e − 15**	7.2068e − 15
80	30	2000	1.3323e − 14	**9.2726e − 15**	1.2825e − 14

Table 5: Test results of Rastrigin function.

M	Dimension	Iteration	QPSO	MQPSO	IQPSO
20	10	1000	3.9177	5.2533	**3.4057**
20	20	1500	14.6960	22.1470	**14.1137**
20	30	2000	30.0435	34.7817	**28.9520**
40	10	1000	**2.6774**	3.8467	2.7584
40	20	1500	10.3371	11.9833	**9.0176**
40	30	2000	19.9741	26.3297	**18.3777**
80	10	1000	1.8968	2.5669	**1.6226**
80	20	1500	7.6702	11.4880	**7.0007**
80	30	2000	15.5625	17.6602	**14.5163**

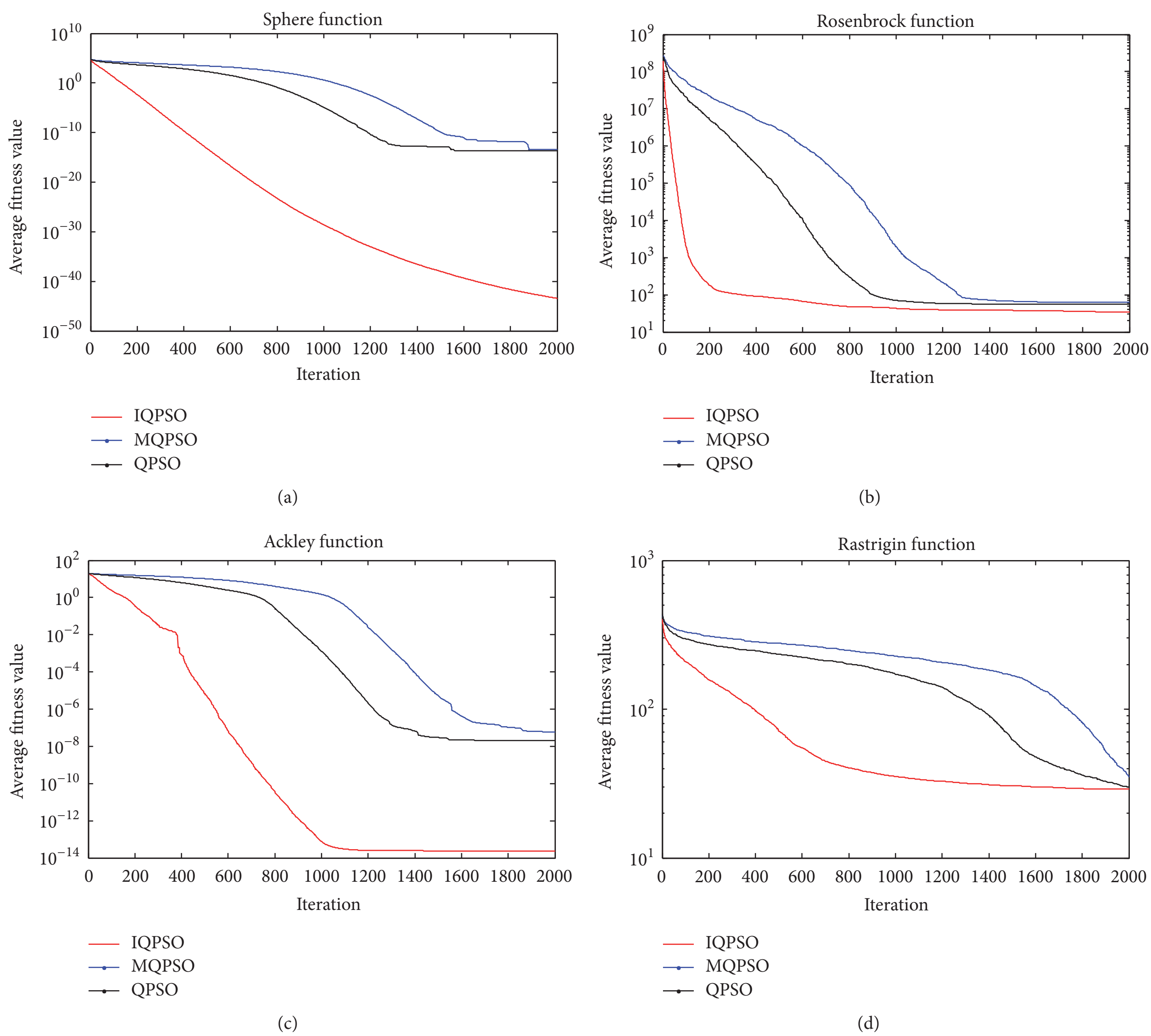

Figure 3: Convergence process of different algorithms on benchmark functions (the average fitness values are represented in log scale). (a) Sphere function; (b) Rosenbrock function; (c) Ackley function; (d) Rastrigin function.

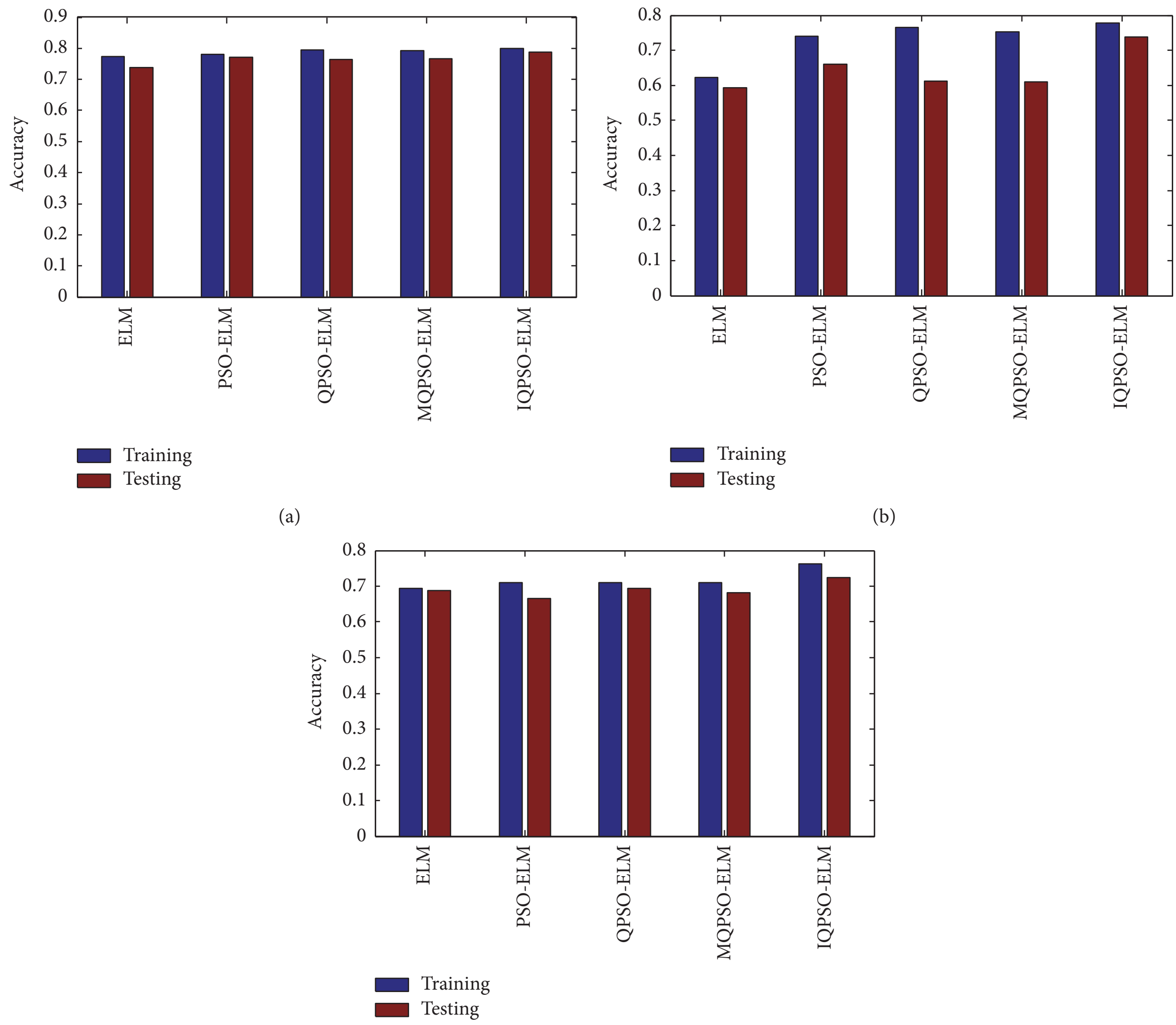

Figure 4: The performance comparison of different methods on the real-world datasets: (a) Diabetes; (b) Liver-disorders; (c) Auto-MPG.

that the convergence speed of the proposed method is much faster than that of the other two methods. Then we know that the proposed method has strong ability of global optimization and could generate better solutions.

4.2. The Classification Performance of IQPSO-ELM on Real-World Datasets. In this section, some real-world datasets such as Diabetes, Liver-disorders, and Auto-MPG are used to test the proposed algorithm. What needs to be explained is that Diabetes dataset is a dataset about diabetes, which includes patient physiological data and disease progression after one year; the dataset is divided into 2 classes, with a total of 768 samples; each sample has 8 kinds of attribute values, that is, the number of features. Liver-disorders dataset is a medical research database donated by Richard S. Forsyth. There are 345 samples in this dataset, each sample has 6 attribute values, and the dataset is divided into 2 categories. Auto-MPG dataset, which concerns the city-cycle fuel consumption, is taken from the StatLib library which is maintained at Carnegie Mellon University. There are 398 samples in Auto-MPG dataset and each sample has 7 attribute values. The Auto-MPG dataset is divided into 3 classes. The detailed description of the three datasets is listed in Table 6. It is important to note that the training samples, test samples, and validation samples are randomly generated according to the number listed in Table 6. Figure 4 shows the performance comparison of five algorithms, that is, ELM, PSO-ELM, QPSO-ELM, MQPSO-ELM, and IQPSO-ELM. We should know that the number of hidden nodes of the five algorithms is 10 and the maximum number of iterations of evolutionary ELM algorithms is 100. In order to avoid accidental results, the algorithms run 50 times, respectively, and then calculate the average value of results.

Table 6: The description for the used datasets.

Datasets	Attributes	Classes	Number of samples		
			Training	Test	Validation
Diabetes	8	2	192	288	288
Liver-disorders	6	2	169	88	88
Auto-MPG	7	3	200	99	99

Table 7: The description for the target.

Target	Height	(Bottom) diameter	Cone angle
Conical target	1.2 m	0.52 m	31.8°
Cylindrical target	1 m	0.5 m	-
Cone-like target	0.91 m	0.325 m	16°
Spherical target	-	1 m	-

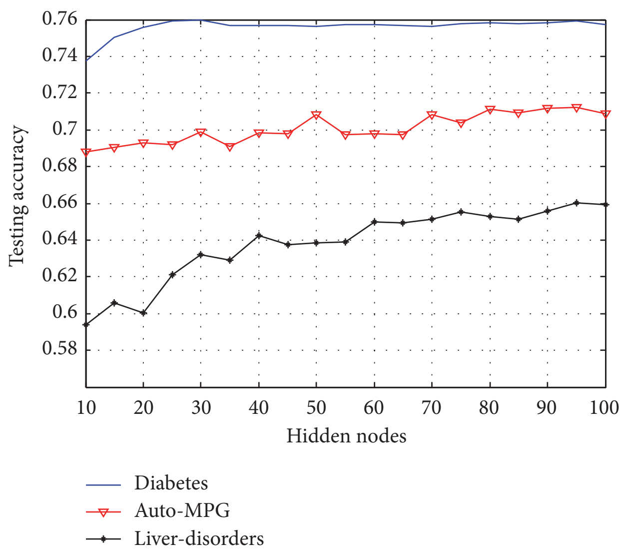

Figure 5: Test results of ELM on three different datasets with different hidden nodes.

Table 8: Performance comparison of different methods.

Methods	Training		Testing	
	Time (s)	Accuracy	Time (s)	Accuracy
BP	0.2040	0.9950	0.0148	0.9915
SVM	53.1855	1	3.5684	0.9966
ELM	0.0016	0.9609	0.0019	0.9437
PSO-ELM	254.3718	0.9884	0.0017	0.9864
QPSO-ELM	130.5861	0.9919	0.0011	0.9803
MQPSO-ELM	128.9230	0.9927	0.0011	0.9779
IQPSO-ELM	133.1459	0.9945	0.0012	0.9921

From Figure 4, we know that, in the aspect of training and testing accuracy, the proposed method can achieve higher accuracy on the given three datasets, which shows the effectiveness of the proposed method. Figure 5 shows the testing accuracies of ELM with varying hidden nodes on the three datasets. As can be seen from Figure 5, with the increase of the number of hidden nodes, the testing accuracy of ELM has been improved. For the Diabetes dataset, when the hidden layer node is set to 30, the test accuracy is up to 0.7598, for the Liver-disorders dataset, when the number of hidden nodes is 95, the test accuracy is 0.6601, and for the Auto-MPG dataset, when the number of hidden layer nodes is 95, the test accuracy is up to 0.7123. Although ELM can greatly improve the classification speed, in order to achieve better performance, ELM needs more hidden nodes which may increase the complexity of the network. The proposed method can use a simple network to obtain a good classification result.

4.3. The Classification Performance of IQPSO-ELM on Radar Data. In this section, simulated data and darkroom measured data are utilized to verify the validity of IQPSO-ELM. The performance comparison of BP, SVM, ELM, and evolutionary ELM including PSO-ELM, QPSO-ELM, MQPSO-ELM, and IQPSO-ELM in this experiment is given in Tables 8 and 9. It should be noted that the number of hidden nodes of evolutionary ELM is set to 10, the swarm size is 20, and the maximum number of iterations is about 100. The number of hidden nodes of BP is set to 10, the activation functions are "*logsig*" and "*purelin*," the training function is "*trainlm*," and the learning rate is 0.01. The kernel function of SVM is "*guass*" and the penalty factor is 0.2. The number of hidden nodes of ELM is 10 and the hidden layer activation function is "*sigmoid*."

4.3.1. Simulated Data. Radar target classification technology is of great significance in both military and civil aspects. The classification accuracy and real-time performance are particularly important. Previously widely used classification methods, such as BP and SVM, have the problems of low classification accuracy or long time consumption. It is hard for them to be excellent in two aspects of the real-time performance and accuracy. To solve this issue, we propose IQPSO-ELM algorithm, which not only makes full use of the advantages of fast learning speed and good generalization performance of ELM, but also utilizes the improved QPSO to obtain the appropriate input weights and hidden layer biases. The proposed method is able to solve the above problem very well.

In this section, we utilize the simulated data to validate the proposed algorithm. The simulated radar targets include conical target, cylindrical target, cone-like target, and spherical target. The size of each target model is shown in Table 7. It is assumed that the radar observation time is from 100 to 600 s. The selected target features include RCS mean, RCS variance, scatter centers, and Micro-Motion period. According to the prior knowledge and the related work [1, 2, 36], the data is simulated by a certain random error added to a set of truth values. In the simulated data, the RCS mean value of the target is set to −5 dB, −4 dB, −4.5 dB, and −2 dB, respectively, and

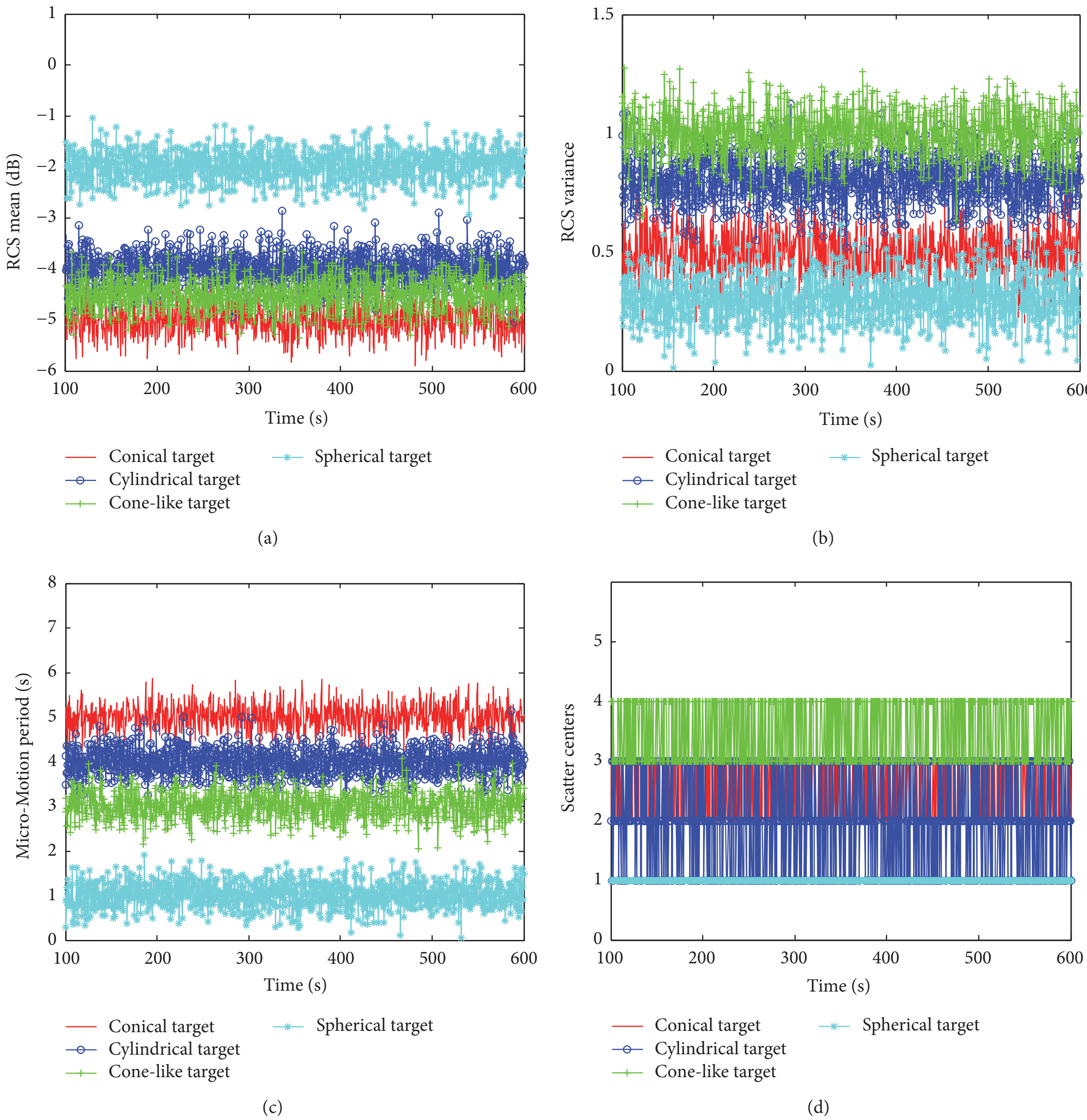

Figure 6: Simulated data of radar target features: (a) RCS mean; (b) RCS variance; (c) Micro-Motion period; (d) scatter centers.

the relative error is 0.3, the RCS variance is set to 0.5, 0.8, 1.0, and 0.3, respectively, and the relative error is 0.1, the Micro-Motion period is set to 5 s, 4 s, 3 s, and 1 s respectively, and the relative error is 0.3, and the number of scattering centers is set to 2, 2, 3, and 1, respectively, and the relative error is 1. The simulated data is shown in Figure 6.

From Table 8, we know that the BP, SVM, ELM, and the evolutionary ELM including PSO-ELM, QPSO-ELM, MQPSO-ELM, and IQPSO-ELM algorithms have good effect on radar target classification. In the training phase, the ELM takes the least time and shows its fast learning ability. The evolutionary ELM algorithms take more time than other algorithms; that is because the evolutionary ELM algorithms need to spend some time on optimizing the input weights and hidden biases. In the testing phase, although the accuracy of the proposed algorithm is slightly lower than that of SVM, the test time is greatly reduced. Compared with ELM, although the test times of the two algorithms are not much different, the accuracy of proposed method is higher. From Figure 7, we know that, in order to achieve better classification performance, ELM needs more hidden layer nodes, which results in complex network structure.

Table 9: Performance comparison of different methods.

Methods	Training		Testing	
	Time (s)	Accuracy	Time (s)	Accuracy
BP	0.0958	0.9150	0.0134	0.9020
SVM	0.8336	1	0.0566	0.9984
ELM	6.98e − 4	0.8752	3.22e − 4	0.7284
PSO-ELM	6.4222	0.9984	3.12e − 4	0.9733
QPSO-ELM	3.4785	0.9905	3.33e − 4	0.9716
MQPSO-ELM	3.4429	0.9879	3.12e − 4	0.9681
IQPSO-ELM	3.5496	0.9935	3.24e − 4	0.9804

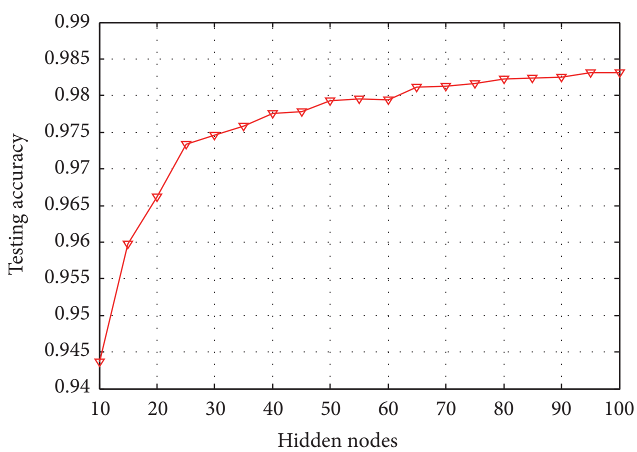

Figure 7: Accuracy of ELM in different hidden nodes.

The experimental results show that the proposed algorithm can meet the requirements of real-time performance and accuracy in radar target classification.

4.3.2. Darkroom Measured Data. In this section, darkroom measured data is used to verity the validity of the proposed method. The radar targets include the conical target, cylindrical target, and cone-like target. The features of radar targets are shown in Figure 8, including RCS mean, RCS variance, Micro-Motion period, and Micro-Motion amplitude. Compared with the simulated data, due to the scene complexity, the darkroom measured data has the characteristics of large dynamic range and low stability.

According to Table 9, the evolutionary ELM methods take more time in the training process; that is because the evolutionary ELM algorithms need to spend some time on optimizing the input weights and hidden biases. In the testing process, the accuracy of the proposed method is improved compared with BP and ELM; although not as good as SVM, the proposed method takes less time than SVM. It can be seen from Figure 9 that, in order to get better classification effect, ELM needs more hidden nodes, which makes the network structure more complex. Therefore, considering the two aspects of the real-time and accuracy, the IQPSO-ELM method is better.

5. Conclusions

In this paper, we proposed a novel evolutionary extreme learning machine based on improved quantum-behaved particle swarm optimization. Further, we introduced it to radar target classification. The proposed method not only makes full use of the advantages of fast learning speed and good generalization performance of ELM, but also utilizes the improved QPSO to obtain the appropriate input weights and hidden layer biases, which is able to solve the problem that the ELM needs more hidden nodes to get better classification performance. The performance of the IQPSO method is evaluated on the well-known benchmark functions. The experimental results show that the proposed method can not only achieve the best solutions, but also converge to the optimal solution faster than other methods. Moreover, the results of the experiment on the real-world datasets show that the proposed IQPSO-ELM method can achieve good performance. Finally, some experiments on radar target classification verify the effectiveness of IQPSO-ELM. Although the classification accuracy of the proposed method is a little lower than that of SVM, it runs much faster than SVM. The experimental results show that the proposed method is more cost-efficient than other traditional radar target classification methods. We are absolutely convinced that the work presented in this paper is extremely significant for radar target classification. We will further optimize our proposed method and expand the scope of its application in the near future.

Acronyms

ELM:	Extreme learning machine
PSO:	Particle swarm optimization
QPSO:	Quantum-behaved particle swarm optimization
IQPSO:	Improved quantum-behaved particle swarm optimization
D-S:	Dempster-Shafer
SVM:	Support vector machine

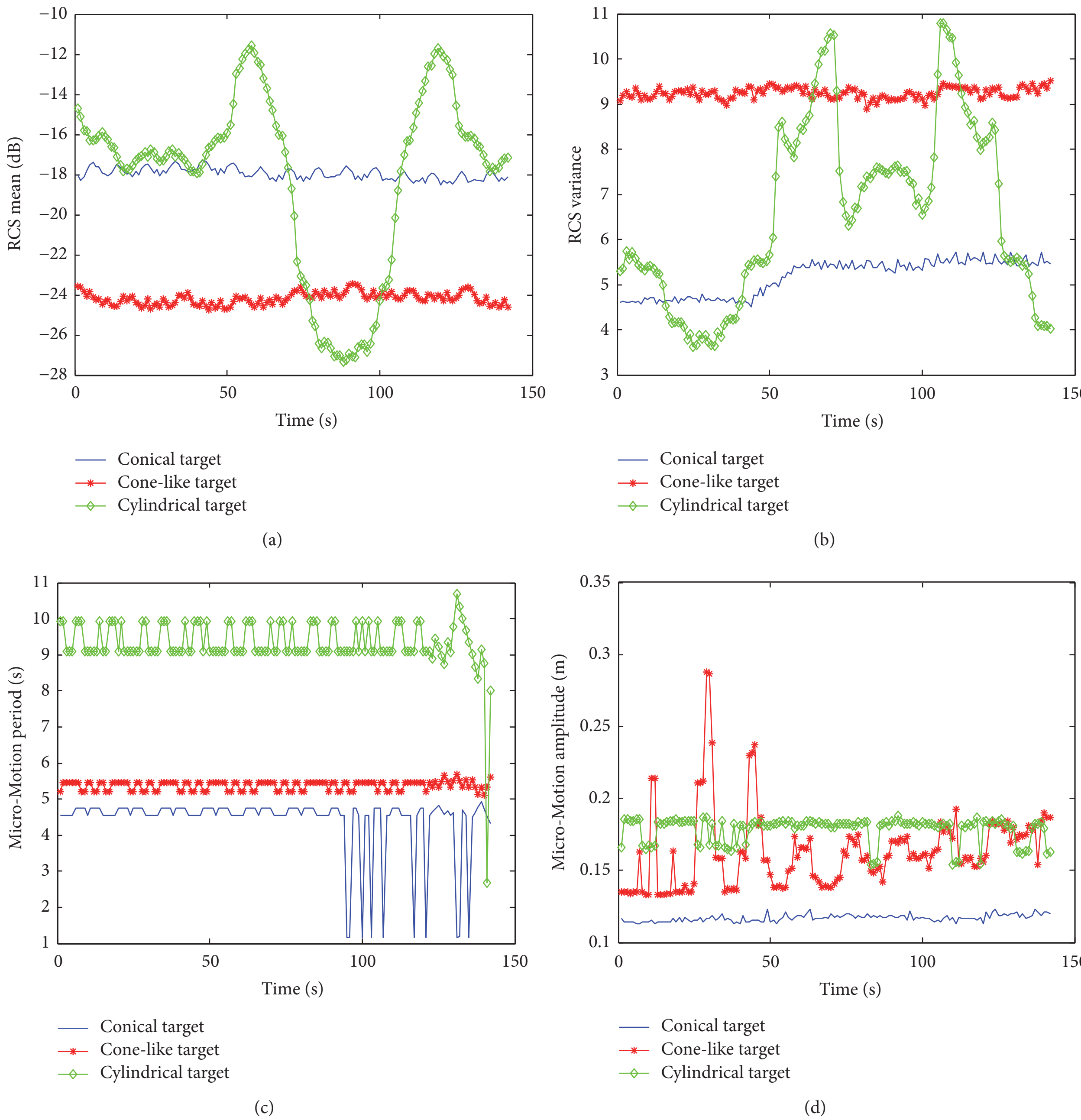

FIGURE 8: Darkroom measured data of radar target features: (a) RCS mean; (b) RCS variance; (c) Micro-Motion period; (d) Micro-Motion amplitude.

BP: Back propagation
SLFNs: Single-hidden-layer feedforward neural networks
C-ELM: Complex ELM
OS-ELM: Online sequential ELM
Ada-ELM: Adaptive ensemble model of ELM
I_ELM: Incremental ELM
P_ELM: Pruned ELM
SaELM: Self-adaptive ELM
MR: Manifold regularization
SBELM: Sparse Bayesian ELM
CLPSO: Comprehensive learning particle swarm optimizer
EWQPSO: Enhanced weighted quantum PSO
RLQPSO: Random local optimized QPSO
MQPSO: Modified QPSO.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

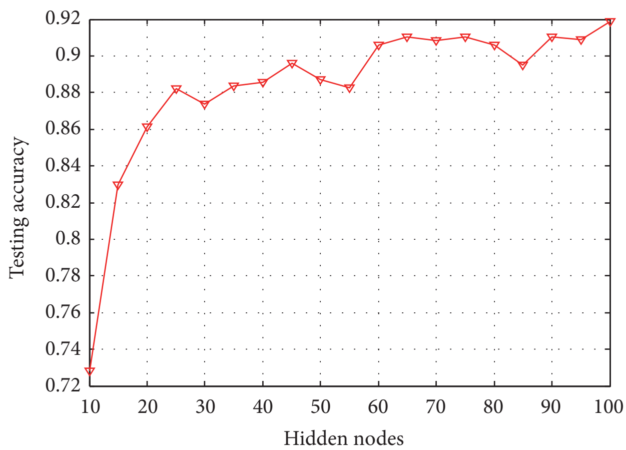

Figure 9: Accuracy of ELM in different hidden nodes.

Acknowledgments

The work in this paper has been supported by the National Science Foundation of China (61422114 and 61501481) and the Natural Science Fund for Distinguished Young Scholars of Hunan Province under Grant no. 2015JJ1003.

References

[1] Y. Liu, D. Zhu, X. Li, and Z. Zhuang, "Micromotion characteristic acquisition based on wideband radar phase," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 6, pp. 3650–3657, 2014.

[2] S. H. Zhang, Y. X. Liu, and X. Li, "Autofocusing for sparse aperture isar imaging based on joint constraint of sparsity and minimum entropy," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 3, pp. 998–1011, 2017.

[3] T.-T. Wong and L.-H. Chang, "Individual attribute prior setting methods for nave Bayesian classifiers," *Pattern Recognition*, vol. 44, no. 5, pp. 1041–1047, 2011.

[4] T. Denoeux, "A neural network classifier based on Dempster-Shafer theory," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 30, no. 2, pp. 131–150, 2000.

[5] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[6] J. Liu, N. Fang, Y. J. Xie, and B. F. Wang, "Radar target classification using support vector machine and subspace methods," *IET Radar, Sonar & Navigation*, vol. 9, no. 6, pp. 632–640, 2015.

[7] R. Mou, Q. Chen, and M. Huang, "An improved BP neural network and its application," in *Proceedings of the 4th International Conference on Computational and Information Sciences, ICCIS*, pp. 477–480, Chongqing, China, August 2012.

[8] 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.

[9] Y. Zeng, X. Xu, D. Shen, Y. Fang, and Z. Xiao, "Traffic sign recognition using kernel extreme learning machines with deep perceptual features," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–7, 2016.

[10] W. Zong and G.-B. Huang, "Face recognition based on extreme learning machine," *Neurocomputing*, vol. 74, no. 16, pp. 2541–2551, 2011.

[11] A. Iosifidis, A. Tefas, and I. Pitas, "Minimum class variance extreme learning machine for human action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 11, pp. 1968–1979, 2013.

[12] N. Liu and H. Wang, "Evolutionary extreme learning machine and its application to image analysis," *Journal of Signal Processing Systems*, vol. 73, pp. 1–9, 2013.

[13] 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.

[14] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: a review," *Neural Networks*, vol. 61, pp. 32–48, 2015.

[15] M.-B. Li, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "Fully complex extreme learning machine," *Neurocomputing*, vol. 68, no. 1-4, pp. 306–314, 2005.

[16] N. Liang, G. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 17, no. 6, pp. 1411–1423, 2006.

[17] H. Wang, W. Fan, F. Sun, and X. Qian, "An adaptive ensemble model of extreme learning machine for time series prediction," in *Proceedings of the 12th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP*, pp. 80–85, Chengdu, China, December 2015.

[18] G. Huang, L. Chen, and C. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 17, no. 4, pp. 879–892, 2006.

[19] H.-J. Rong, Y.-S. Ong, A.-H. Tan, and Z. Zhu, "A fast pruned-extreme learning machine for classification problem," *Neurocomputing*, vol. 72, no. 1–3, pp. 359–366, 2008.

[20] G. G. Wang, M. Lu, Y. Q. Dong, and X. J. Zhao, "Self-adaptive extreme learning machine," *Neural Comput Applic*, vol. 27, pp. 291–303, 2016.

[21] B. Liu, S. X. Xia, F. R. Meng, and Y. Zhou, "Manifold regularized extreme learning machine," *Neural Comput Applic*, vol. 27, pp. 255–269, 2016.

[22] J. Luo, C.-M. Vong, and P.-K. Wong, "Sparse bayesian extreme learning machine for multi-classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 4, pp. 836–843, 2014.

[23] E. M. N. Figueiredo and T. B. Ludermir, "Investigating the use of alternative topologies on performance of the PSO-ELM," *Neurocomputing*, vol. 127, pp. 4–12, 2014.

[24] J. Sun, W. Fang, X. Wu, V. Palade, and W. Xu, "Quantum-behaved particle swarm optimization: analysis of individual particle behavior and parameter selection," *Evolutionary Computation*, vol. 20, no. 3, pp. 349–393, 2012.

[25] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[26] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, vol. 1, pp. 325–331, June 2004.

[27] P. Bia, D. Caratelli, L. Mescia, and J. Gielis, "Analysis and synthesis of supershaped dielectric lens antennas," *IET Microwaves, Antennas & Propagation*, vol. 9, no. 14, pp. 1497–1504, 2015.

[28] C. Zhang, Y. Xie, D. Liu, and L. Wang, "Fast threshold image segmentation based on 2D fuzzy Fisher and random local optimized QPSO," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1355–1362, 2017.

[29] Z. Yang, X. Wen, and Z. Wang, "QPSO-ELM: An evolutionary extreme learning machine based on quantum-behaved particle swarm optimization," in *Proceedings of the 7th International Conference on Advanced Computational Intelligence, ICACI*, pp. 69–72, Fujian, China, March 2015.

[30] C. Peng, J. Yan, S. Duan, L. Wang, P. Jia, and S. Zhang, "Enhancing electronic nose performance based on a novel QPSO-KELM model," *Sensors*, vol. 16, no. 4, article no. 520, 2016.

[31] X. Yang, S. Pang, W. Shen, X. Lin, K. Jiang, and Y. Wang, "Aero engine fault diagnosis using an optimized extreme learning machine," *International Journal of Aerospace Engineering*, vol. 2016, Article ID 7892875, 10 pages, 2016.

[32] C. Jin and S.-W. Jin, "Automatic image annotation using feature selection based on improving quantum particle swarm optimization," *Signal Processing*, vol. 109, pp. 172–181, 2015.

[33] J. Sun, X. Wu, V. Palade, W. Fang, C.-H. Lai, and W. Xu, "Convergence analysis and improvements of quantum-behaved particle swarm optimization," *Information Sciences*, vol. 193, pp. 81–103, 2012.

[34] M. Xi, J. Sun, and W. Xu, "An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position," *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 751–759, 2008.

[35] J. Sun, C. H. Lei, W. B. Xu, and Z. L. Chai, "A novel and more efficient search strategy of quantum-behaved particle swarm optimization," in *Proceedings of 8th International Conference on Adaptive and Natural Computing Algorithms ICANNGA*, vol. 4431, pp. 394–403, Warsaw, Poland, Apr. 2007.

[36] M. Vespe, C. J. Baker, and H. D. Griffiths, "Radar target classification using multiple perspectives," *IET Radar, Sonar & Navigation*, vol. 1, no. 4, pp. 300–307, 2007.