Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 9726529, 8 pages
https://doi.org/10.1155/2017/9726529

Research Article

Aeroengine Fault Diagnosis Using Optimized Elman Neural Network

Jun Pi, Jiangbo Huang, and Long Ma

Sino-European Institute of Aviation Engineering, Civil Aviation University of China, Tianjin 300300, China

Correspondence should be addressed to Jiangbo Huang; 1634401546@qq.com

Received 8 June 2017; Accepted 22 November 2017; Published 19 December 2017

Academic Editor: Alessandro Gasparetto

A new Elman Neural Network (ENN) optimized by quantum-behaved adaptive particle swarm optimization (QAPSO) is introduced in this paper. According to the root mean square error, QAPSO is used to select the best weights and thresholds of the ENN in training samples. The optimized neural network is applied to aeroengine fault diagnosis and is compared with other optimized ENN, original ENN, BP, and Support Vector Machine (SVM) methods. The results show that the QAPSO-ENN is more accurate and reliable in the aeroengine fault diagnosis than the conventional neural network and other ENN methods; QAPSO-ENN has great diagnostic ability in small samples.

1. Introduction

The operational safety, economy, and reliability of aeroengine are problems of primary concern for airline companies [1]. The fault diagnosis plays a key role in enhancing the working safety and reliability and reducing the operating cost of aircraft engine. Therefore, accurate aeroengine fault identification in advance provides a sufficient decision-making time for the airline maintenance plan development and avoids excess maintenance and inadequate maintenance. At present, the commonly used aeroengine fault diagnosis methods include expert system, fuzzy reasoning, pattern recognition, and intelligent computation.

In recent years, intelligent computing methods, especially neural network algorithms, have been rapidly developed in image processing, control, and other fields. The neural network is successfully introduced into the fault diagnosis by Zedda and Singh [2]; the BP neural network belongs to the static networks which make the dynamic network of the original aeroengine state monitoring time become a static model on the space. So it is not suitable for aeroengine fault diagnosis; Elman Neural Network (ENN) [3], a dynamic recursive network, has the characteristics of fast training speed, simple structure, and high prediction precision compared with BP neural network and has been widely studied in the literature. Elman Neural Network was used by Gao et al. to identify the dynamical systems [4]. In order to improve the prediction accuracy of network traffic, Wang et al. proposed a modified Elman Neural Network for the network system [5]. Lin and Hong present an improved Elman Neural Network-(IENN-) based algorithm for optimal wind-energy control with maximum power point tracking [6]. However, it also have the problems of falling into the local minimum value and the convergence rate slowing down when dealing with small samples, high dimension, and deteriorated network performance. In addition, if the initial weight and threshold selection are not appropriate, ENN convergence becomes difficult and cannot get the desired prediction effect.

Particle swarm optimization (PSO) [7, 8], an intelligent optimization algorithm, has the characteristics of fast convergence and high precision and has been applied in many fields [9] such as being used for optimal location of flexible AC transmission system devices considering cost of installation and system loadability [10] and solving the economic dispatch (ED) problem of power system [11]. However, PSO also has the problem that it easily falls into the local minimum, which leads to the optimized ENN by PSO not as good as the ordinary ENN in solving the small sample problem.

In order to solve the problem that the PSO extremely easily falls into the minimum value, the quantum-behaved adaptive particle swarm optimization (QAPSO) method based on the ENN combining the adaptive particle swarm

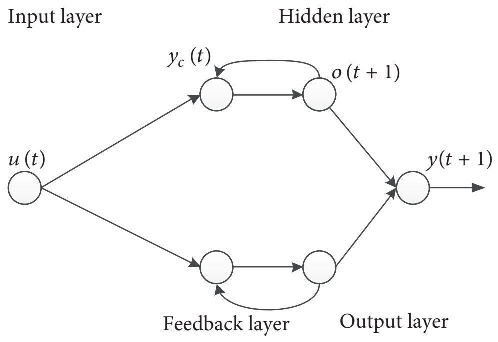

Figure 1: Elman Neural Network structure diagram.

optimization algorithm [12] and quantum-behaved particle swarm optimization algorithm proposed by Sun et al. [13, 14] is proposed in this paper and is utilized to diagnose the aeroengine faults. The diagnosis results are reliable and suitable [15, 16].

2. Brief of Elman Neural Network

ENN was proposed by Elman in 1990 and its structure was shown in Figure 1.

As shown in Figure 1, $u(t)$ is the network input on the time series, $y_c(t)$ is the output of the feedback layer, $y(t)$ is the network output, and $o(t)$ is the hidden layer output; the network calculation process is as follows:

$$\begin{aligned} x_0(t+1) &= Hy_c(t) + Wu(t) + \theta \\ y_c(t) &= o(t-1) = f_1(x_0(t-1)) \\ y(t) &= f_2(Ay_c(t) + \varphi), \end{aligned} \tag{1}$$

where $x_0(t)$ is the input of the hidden layer, W, H, A are the connection weight matrix of the input layer to the hidden layer, the feedback layer to the hidden layer, and the hidden layer to the output layer; $f_1(\bullet)$ And $f_2(\bullet)$ are the transfer function of the hidden layer and the output layer.

Whether the network training has stopped or not, the judgment error function is

$$E = \sum_{t=1}^{n} \|y(t) - d(t)\|_2^2, \tag{2}$$

where $d(t)$ is the expected output of the neural network; $y(t)$ is the actual output of the neural network; $\|\bullet\|_2$ represents the Frobenius norm of the vector.

3. Brief of Particle Swarm Optimization

In the N-dimensional search space, the particles fly at a certain speed, $\mathbf{X}_i = (x_{i1}, x_{i2}, \ldots, x_{iN})$ is the position of the ith particle, $\mathbf{V}_i = (v_{i1}, v_{i2}, \ldots, v_{iN})$ is the ith particle current flight speed, and $\mathbf{pbest_i} = (pbest_{i1}, pbest_{i2}, \ldots, pbest_{iN})$ is the optimal position experienced by the ith particle, which can be considered the individual optimal position. $\mathbf{Gbest_j} = (zbest_{j1}, zbest_{j2}, \ldots, zbest_{jN})$ is the optimal position searched by the entire particle group, which can be considered as the global optimal position. The speed and position updating equation of each particle are

$$\begin{aligned} v_{ij}(t+1) &= v_{ij}(t) + c_1 r_2 (pbest_{in}(t) - x_{in}(t)) \\ &\quad + c_1 r_2 (gbest_{jn}(t) - x_{in}) \\ x_{in}(t+1) &= x_{in}(t) + v_{in}(t+1), \end{aligned} \tag{3}$$

where $i = 1, 2, \ldots, M$; $n = 1, 2, \ldots, N$; c_1 and c_2 are the acceleration factor or called the learning factor; r_1 and r_2 are the random number of $[0, 1]$.

The basic PSO algorithm is simple to use and has fast searching speed, but it also easily falls into the local minimum and convergence precision problems [17], so it is necessary to improve it.

4. Brief of Quantum-Behaved Adaptive Particle Swarm Optimization

4.1. The Initial Position of the Quantum Particle Group Is Set. The quantum bits are used to replace the particles in the standard particle group, where the positional state of the quantum bits consists of $|0\rangle$ and $|1\rangle$; the quantum bits can be depicted by the following formula.

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{4}$$

where α and β meet the following relationship:

$$|\alpha|^2 + |\beta|^2 = 1, \tag{5}$$

Therefore, we consider that the position of x^α is the position of $|0\rangle$ state, where x^β is the position in the $|1\rangle$ state; the position of the particle at the conventional x at the quantum bit is expressed as

$$x = x^\alpha + x^\beta, \tag{6}$$

The initial position of the standard particle group is transformed into the initial position of the quantum particle group as follows:

$$X_i = \begin{bmatrix} x_{i1}^\alpha & x_{i2}^\alpha & \cdots & x_{ij}^\alpha & \cdots & x_{iN}^\alpha \\ x_{i1}^\beta & x_{i2}^\beta & \cdots & x_{ij}^\beta & \cdots & x_{iN}^\beta \end{bmatrix}. \tag{7}$$

where $i = 1, 2, \ldots, M$; $j = 1, 2, \ldots, N$; N is the dimension of the search space; M is the population size of the particle swarm.

4.2. Particle Update Process. For the ith particle, the average distance and rate with the other particles [18] are

$$\begin{aligned} D_{ix} &= \frac{1}{M-1} \sum_{j=1, j\neq i}^{M} \sqrt{\sum_{k=1}^{N} (x_{ik} - x_{jk})^2} \\ D_{iv} &= \frac{1}{M-1} \sum_{j=1, j\neq i}^{M} \sqrt{\sum_{k=1}^{N} (v_{ik} - v_{jk})^2}. \end{aligned} \tag{8}$$

The particle trajectory difference was determined by the average distance and the average rate and expressed as

$$D_{ic} = D_{ix} + \rho_{X_iV_i} * D_{iv} \tag{9}$$

$$\rho_{X_iV_i} = \frac{E\left(X_iV_i\right) - E\left(X_i\right)E\left(V_i\right)}{\sqrt{E\left(X^2{}_i\right) - E^2\left(X_i\right)} \bullet \sqrt{E\left(X^2{}_i\right) - E^2\left(X_i\right)}} = \frac{\sum_{j=1}^{N} x_{ij}v_{ij} - (1/N)\sum_{j=1}^{N} x_{ij}\sum_{j=1}^{N} v_{ij}}{\sqrt{\sum_{j=1}^{N} x_{ij}^2 - (1/N)\left(\sum_{j=1}^{N} x_{ij}\right)^2} \bullet \sqrt{\sum_{j=1}^{N} v_{ij}^2 - (1/N)\left(\sum_{j=1}^{N} v_{ij}\right)^2}}. \tag{10}$$

where $\rho_{X_iV_i}$ is the correlation coefficient between x and v Pearson.

According to the above formula to calculate D_{ic}, $D_{c\,\mathrm{min}}$ is the largest particle trajectory difference, $D_{c\,\mathrm{max}}$ is the smallest recorded one, and D_{cg} is the optimal particle trajectory difference; the particle trajectory evolution factor was expressed as

$$f_c = \frac{D_{cg} - D_{c\,\mathrm{min}}}{D_{c\,\mathrm{max}} - D_{c\,\mathrm{min}}} \in [0,1]. \tag{11}$$

4.3. Inertial Weight of the Adaptive Dynamic Control. Because of the uneven distribution of the particle itself and the global environment, particle's cognitive ability and cognition of global environment should be allocated dynamically in the process of particle iteration update. On this basis, the following dynamic weight method is proposed to keep the dynamic balance between global optimization and local optimization.

$$w_c\left(f_c\right) = \frac{1}{1 + 1.5e^{-2.6f_c}} \subset [0.4, 0.9]. \tag{12}$$

4.4. Mutation Operation. In order to enable the particles to jump out of the previous search for the optimal position and continue to expand the search space of continuous reduction in the iterative process, while ensuring that the particles have the population diversity in a larger search space, the idea of variation in genetic algorithm was applied to make the position of quantum bits in $|0\rangle$ and $|1\rangle$ status exchanged by controlled gate. Controlled gate definition [19] is

$$C(k) = \begin{bmatrix} \cos\left(\dfrac{k\pi}{2}\right) & \sin\left(\dfrac{k\pi}{2}\right) \\ \sin\left(\dfrac{k\pi}{2}\right) & \cos\left(\dfrac{k\pi}{2}\right) \end{bmatrix}. \tag{13}$$

Controlled doors can be divided into two categories:

(1) $k = 1, |\phi\rangle$ completely flipped

(2) $k = 0, |\phi\rangle$ not flipping.

$[x_{ip}^{\alpha}, x_{ip}^{\beta}]$ is defined as the ith quantum bit x_{gi} in the optimal position of the global optimal particle. The variation procedure is achieved by the following equation:

$$\begin{aligned} C(k) * x_{gi} = C(1) * \begin{bmatrix} x_{ip}^{\alpha} \\ x_{ip}^{\beta} \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{ip}^{\alpha} \\ x_{ip}^{\beta} \end{bmatrix} \\ &= \begin{bmatrix} x_{ip}^{\beta} \\ x_{ip}^{\alpha} \end{bmatrix}. \end{aligned} \tag{14}$$

5. Elman Neural Network Optimized by QAPSO

Because the weights and thresholds in Elman Neural Network were calculated by using random weights and thresholds, there might exist a set of unoptimized even unnecessary weights and thresholds. As a result, the Elman Neural Network difficultly gets satisfying test results when it is trained in the small sample.

In this section, the new algorithm named QAPSO-ENN is proposed to solve the problem. QAPSO-ENN calculation flow chart is shown in Figure 2 and it is divided into two parts: the left side of the QAPSO optimization part and the right side of the Elman Neural Network part; the detailed steps of the QAPSO-ENN are as follows.

Step 1. Firstly, generating the particle population randomly, each particle in the population is constituted by a set of weights and thresholds:

$$\begin{aligned} X_i &= \begin{bmatrix} \omega_{11}^{\alpha} \cdots \omega_{HK}^{\alpha}, & b_1^{\alpha} \cdots b_K^{\alpha}, & w_1^{\alpha} \cdots w_K^{\alpha}, & W_{11}^{\alpha} \cdots W_{KP}^{\alpha}, & c_1^{\alpha} \cdots c_P^{\alpha} \\ \omega_{11}^{\beta} \cdots \omega_{HK}^{\beta}, & b_1^{\beta} \cdots b_K^{\beta}, & w_1^{\beta} \cdots w_K^{\beta}, & W_{11}^{\beta} \cdots W_{KP}^{\beta}, & c_1^{\beta} \cdots c_P^{\beta} \end{bmatrix} \\ v_i &= \begin{bmatrix} v_1^{\alpha} \cdots v_i^{\alpha} \cdots v_M^{\alpha} \\ v_1^{\beta} \cdots v_i^{\beta} \cdots v_M^{\beta} \end{bmatrix}, \end{aligned} \tag{15}$$

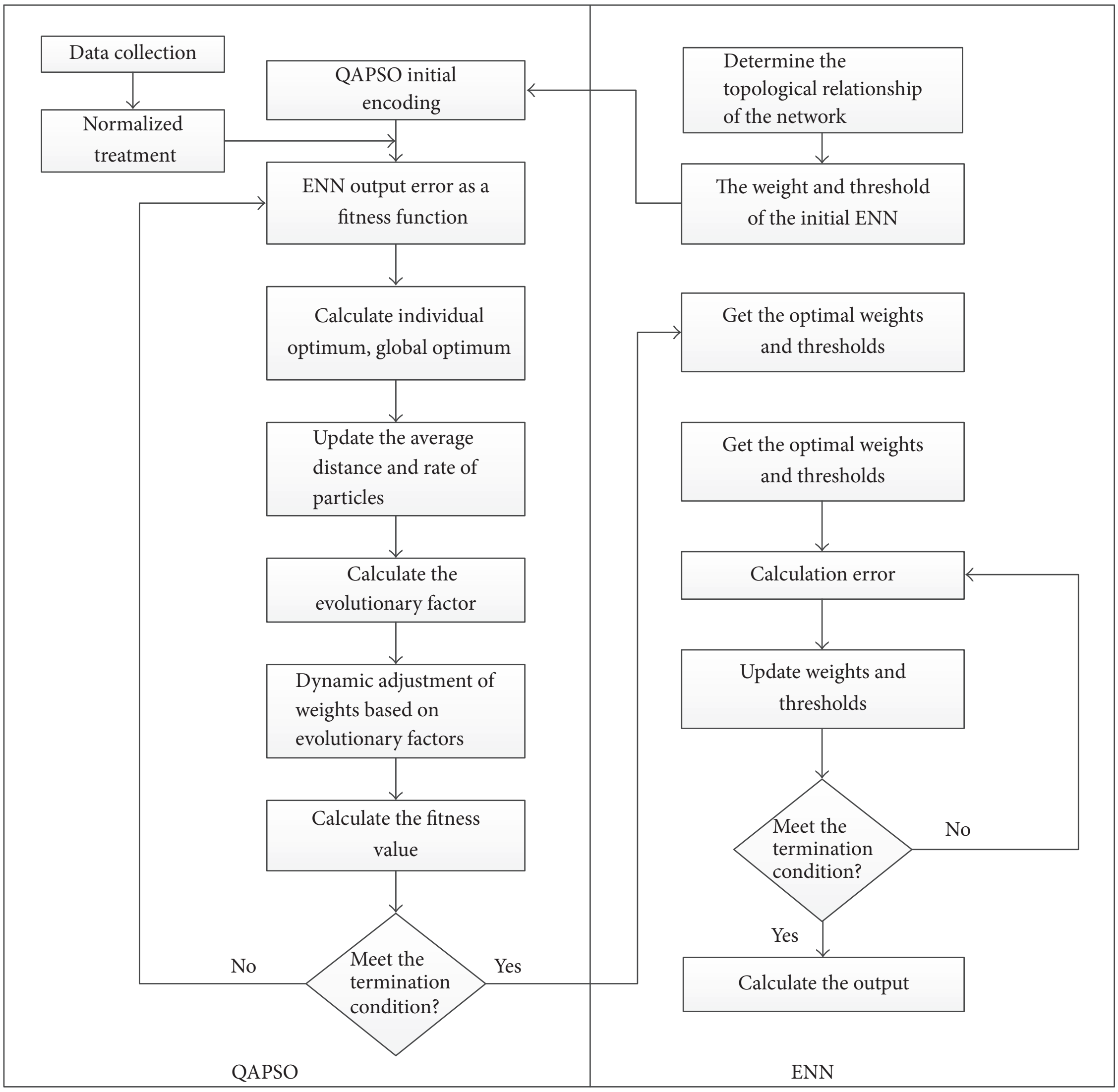
Data collection
Normalized treatment
QAPSO initial encoding
ENN output error as a fitness function
Calculate individual optimum, global optimum
Update the average distance and rate of particles
Calculate the evolutionary factor
Dynamic adjustment of weights based on evolutionary factors
Calculate the fitness value
Meet the termination condition?
No
Yes
QAPSO
Determine the topological relationship of the network
The weight and threshold of the initial ENN
Get the optimal weights and thresholds
Get the optimal weights and thresholds
Calculation error
Update weights and thresholds
Meet the termination condition?
No
Yes
Calculate the output
ENN

Figure 2: QAPSO-ENN flow chart.

where X_i $(i = 1, \ldots, M)$ is the initial position of the particle, M is the particle size of the population, H is the number of input layer nodes, K is the number of hidden layer nodes, M is the number of output layer codes, ω^{α} and ω^{β} are the connection weights between the input layer and hidden layer in the position of $|0\rangle$ and $|1\rangle$ state, b^{α} and b^{β} are the thresholds of hidden layer in the position of $|0\rangle$ and $|1\rangle$ state, w^{α} and w^{β} are the connection weights between the feedback layer and hidden layer in the position of $|0\rangle$ and $|1\rangle$ state, W^{α} and W^{β} are the connection weights between the hidden layer and output layer in the position of $|0\rangle$ and $|1\rangle$ state, b^{α} and b^{β} are the thresholds of output layer in the position of $|0\rangle$ and $|1\rangle$ state, and v^{α} and v^{β} are the speeds of the particle in the position of $|0\rangle$ and $|1\rangle$ state.

Step 2. The difference between the expected output and actual output of ENN is used as the fitness function of QAPSO. The fitness of each particle is calculated according to the fitness function and selects the individual optimum and global optimum.

$$F_i = \sum_{t=1}^{P} \left\| y(t) - d(t) \right\|_2^2, \tag{16}$$

where $d(t)$ is the expected output of the neural network; $y(t)$ is the actual output of the neural network; P is the number of nodes in output layer.

Step 3. The average distance and the average velocity of the particles are calculated according to (9). The maximum average distance and the minimum average distance are selected to calculate the evolution factor by formula (12); then, the weight of the particle group is dynamically regulated according to the calculated evolution factor and formula (13).

Step 4. Particles' new position and velocity are updated according to formula (3). Then the fitness variance is calculated and compared with the set value to determine whether mutation operation is needed.

Table 1: Fault mode and output labels.

Number	Fault models	Label
(1)	2.5 open air valve	1 0 0 0
(2)	2.9 open air valve	0 1 0 1
(3)	3.5 open air valve	0 0 1 0
(4)	3.0 open air valve	0 0 0 1

Finally, we repeat Steps 2–4 until the minimum number of the differences from training Elman Neural Network is reached. Thus the optimized weights and thresholds are obtained, and then the optimized ENN is applied to the aeroengine fault diagnosis.

6. Aeroengine Fault Diagnosis Application

6.1. Fault Model Establishment. In this paper, the GE90 engine fault data of the Boeing 777F aircraft was taken as an example. The four kinds of gas path fault modes are y1-2.5 valve to open, y2-2.9 open gas valve, y3-3.5 open air valve, and y4-3.0 open air valve; then the high-pressure compressor speed (N2), the engine exhaust gas temperature (EGT), and fuel flow (FF), the three performance parameters, are measured by the RD condition monitoring system to diagnose the four kinds of faults [20]. Therefore, the fault diagnosis model is established as follows:

$$Y = f(\mathrm{EGT}, \mathrm{FF}, \mathrm{N2}), \tag{17}$$

where $f(x)$ represents a method, EGT, FF, and N2 are the input of the method, the fault mode is the output of the method, and the output labels are listed in Table 1.

6.2. Normalization. In order to make the equation meet the physical meaning and adapt to the neural network training, the data should be normalized when it enters the engine fault diagnosis model.

$$x = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \tag{18}$$

where $x_{\max}$ and $x_{\min}$ are the maximum and minimum values of a performance parameter (x_i is the ith sample of the one performance parameter); x is the value obtained after normalization.

6.3. The Number of Nodes Selected in Hidden Layer. The number of nodes in the input layer was 3 and the number of nodes in the output layer was 4 by formula (19). However, the number of nodes in hidden layer was not determined by a standard formula; in this paper, the number of nodes in the hidden layer was taken as 5, 6, 7, and 8 based on the empirical formula in [21] for simulation and the results were listed in Table 2.

From Table 2 we can see that the diagnostic accuracy of the network is the highest and the diagnostic performance is the best when the nodes of hidden layer are 7. Therefore, the Elman Neural Network and BP structure established in this paper are determined as (3, 7, 4).

Table 2: Accuracy of different hidden layer codes in aeroengine fault diagnosis.

Code	BP	ENN
5	85.77%	86.18%
6	85.79%	86.76%
7	85.93%	86.97%
8	85.06%	85.97%

The particle dimension was calculated as follows:

$$S = S_{\text{in}} * S_h + S_{\text{out}} * S_h + S_h + S_{\text{out}} + S_c, \tag{19}$$

where S_{in} is the number of nodes in the input layer of the neural network; S_h is the number of nodes in the hidden layer of the neural network. S_{out} is the number of nodes in the output layer of the neural network; S_c is the number of nodes in feedback layer of the neural network.

6.4. Evaluation of the Results. The mean accuracy was calculated as follows:

$$\mathrm{MA} = \frac{1}{T}\sum_{t=1}^{T}\left(1 - \frac{\left|y(t) - d(t)\right|}{d(t)}\right), \tag{20}$$

where MA is the mean accuracy and T is the number of test samples. $y(t)$ is the network prediction value; $d(t)$ is the true value.

7. QAPSO-Elman Algorithm Validation

7.1. Parameter Setting. Aeroengine faults diagnosis is a pattern classification problem. In this section, we apply the QAPSO-ENN on faults diagnosis and compare it with improved PSO optimized ENN(IPSO-ENN), PSO optimized ENN (PSO-ENN), standard ENN, BP, and SVM [22].

All of the six methods were used in aeroengine faults diagnosis by programming in MATLAB R2012a environment running on a PC with 3.2 GHz CPU with 4.0 GB RAM. In the network training, we get 2000 samples where each fault mode has 500 groups and the first 400 groups of each model were used as training samples and the remaining 100 groups were used as test samples. The parameters of each method are set as shown in Tables 3 and 4. The SVM parameters choose default value provided in the LIBSVM 3.12 toolbox.

7.2. Simulation Results of Training Time. It is important to take the time in the actual aeroengine fault diagnosis process. A quick and accurate method can provide more time for the maintenance decision and maintenance plan; therefore, in this section, we train all the methods and count the training times in the different number of training samples and the results are listed in Table 5.

We can see from Table 5 that the training time of the 6 methods is decreasing when the number of training samples decreases. The training time of the three optimized ENN methods is much more than the others, because much of training time of optimized ENN is spent on evaluating all the individuals iteratively.

Table 3: Parameters in the neural network.

Method	Number of nodes in input layer	Number of nodes in hidden layer	Number of nodes in output layer	Error	Epoch
ENN	3	7	4	0.01	10000
BP	3	7	4	0.01	10000

Table 4: Parameters in the particle swarm optimization algorithm.

Method	$x_{\max}/x_{\min}$	$v_{\max}/v_{\min}$	M	Iteration times	$w_{\max}/w_{\min}$	c_1/c_2
PSO	5/−5	0.4/−0.4	67	200	0.9/0.4	2.05/2.05
IPSO	5/−5	0.4/0.4	67	200	0.9/0.4	2.05/2.05
QAPSO	5/−5	0.4/−0.4	67	200	0.9/0.4	2.05/2.05

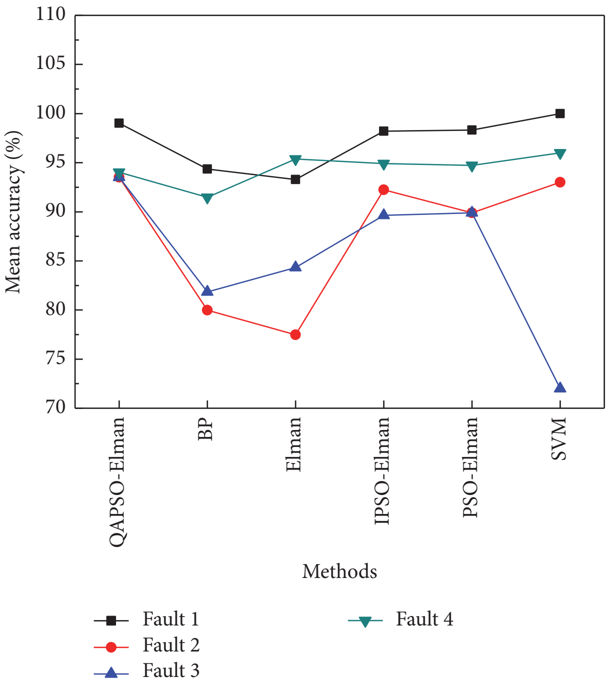

Figure 3: Diagnostic accuracy of four fault modes in the 1600 training samples.

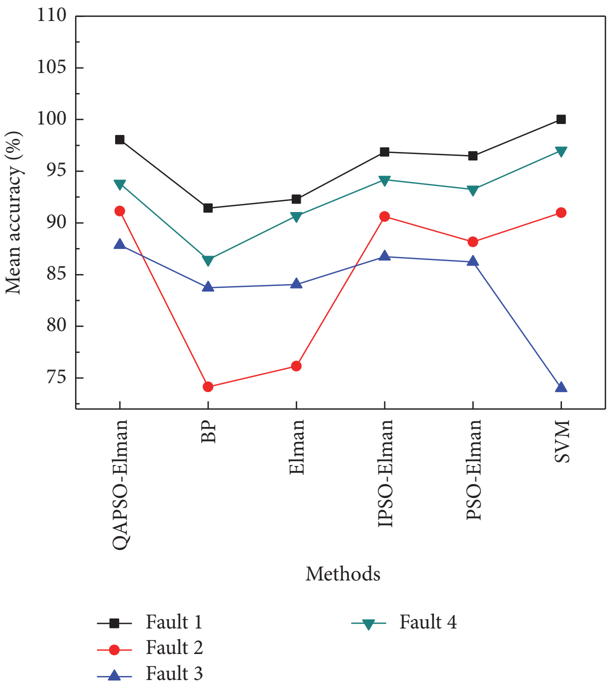

Figure 4: Diagnostic accuracy of four fault modes in the 1200 training samples.

7.3. Simulation Results in Accuracy. In Section 6.2 we train the six methods and study the training times of all the methods in the 1600/1200/800/400 training samples. In this section, the trained methods are used to test the accuracy of each fault mode in different training samples and the results are shown in Figures 3–6.

The performances of the QAPSO-ENN and other methods were compared in 1600/1200/800/400 training samples. Figures 3–6 show each fault mode accuracy of all methods in different training samples. It can be seen from Figure 3 that QAPSO-ENN performed the best results on fault 2 and fault 3, while SVM and QAPSO performed the best on fault 1 and the ENN performed the best on fault 4. From the other figures, we can see that QAPSO-ENN also performs best on fault 2 and fault 3; meanwhile, QAPSO-ENN and SVM also perform the best on fault 1 and QAPSO-ENN and ENN perform the best on fault 4.

In general, the optimized ENN obtained better result than ENN and BP in aeroengine fault diagnosis, which can be confirmed in the four figures; the accuracy of any optimized ENN methods is more accurate than the other three methods. It can be also observed that SVM is better than ENN and BP and fault 3 is more difficult to diagnose than other faults. However, the optimized ENN is more accurate than SVM in fault 3 so we think this result is significantly correlated with the optimized operation for ENN.

In order to verify the diagnostic capability of the six methods in aeroengine and the diagnostic capability in the small samples, the average diagnostic accuracy of each method was calculated in the different training samples and the results ar listed in Table 6.

It can be observed from Table 6 that QAPSO-ENN has much better diagnostic capability than the other 5 methods and obtains the best mean accuracy in 1600/1200/800/400

Table 5: Mean training time on different training samples.

Sample	Training time/s					
	QAPSO-ENN	BP	ENN	IPSO-ENN	PSO-ENN	SVM
1600	222.31	96	60	217.33	218.45	0.160
1200	193.28	87	43	191.37	194.13	0.155
800	153.09	83	34	153.11	148.51	0.151
400	130.20	71	21	129.78	129.53	0.124

Table 6: Average mean accuracy of 6 methods with different numbers of training samples.

Samples	Average mean accuracy/%					
	QAPSO-ENN	BP	ENN	IPSO-ENN	PSO-ENN	SVM
1600	95.02	84.41	87.61	93.74	93.21	90.00
1200	94.26	85.39	87.35	93.43	92.47	89.75
800	94.63	86.40	87.42	93.44	91.71	90.00
400	92.72	83.93	85.79	92.10	91.03	90.25

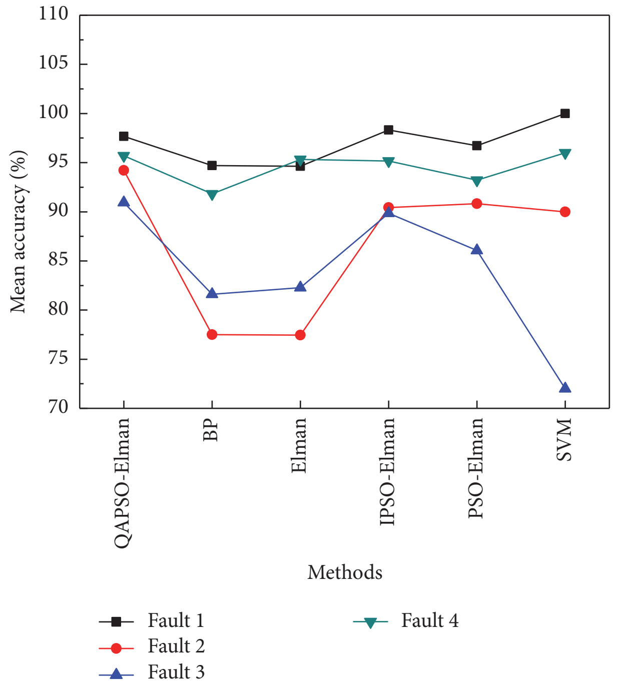

Figure 5: Diagnostic accuracy of four fault modes in the 800 training samples.

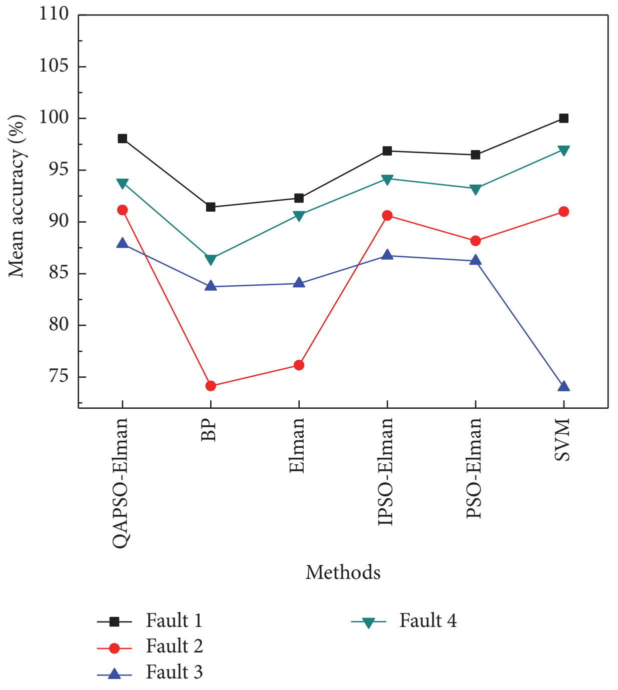

Figure 6: Diagnostic accuracy of four fault modes in the 400 training samples.

samples. IPSO-ENN and QAPSO-ENN are better than PSO-ENN; that is, the QAPSO and the IPSO have overcome the problem that standard PSO easily falls into the local optimum and low convergence accuracy.

8. Conclusions

In this paper, a new Elman Neural Network named QAPSO-ENN was proposed. The proposed method optimizes both the Elman Neural Network weights and thresholds by using QAPSO. In order to validate the performance of the proposed QAPSO-ENN, it was applied to aeroengine fault diagnosis and compared with some other methods. Results show that the QAPSO-ENN is more accurate in most faults than the other optimized ENN methods, SVM, ENN, and BP and shows great advantage by obtaining highest accuracy in the number of 1600 training samples. When the number of training samples decreased, this advantage is also remarkable, which suggests that the QAPSO-ENN is a more reliable and suitable tool for diagnosis of the aeroengine fault.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China and the Civil Aviation Administration of China jointly funded project under Grant no. U1633101.

References

[1] A. K. Sood, C. B. Friedlander, and A. A. Fahs, "Engine Fault Analysis: Part I-Statistical Methods," *IEEE Transactions on Industrial Electronics*, vol. IE-32, no. 4, pp. 294–300, 1985.

[2] S. Osowski, K. Siwek, and T. Markiewicz, "MLP and SVM networks-a comparative study," in *Proceedings of the 6th Nordic Signal Processing Symposium (NORSIG04*, p. 37, 2004.

[3] Y.-C. Cheng, W.-M. Qi, and W.-Y. Cai, "Dynamic properties of Elman and modified Elman neural network," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 637–640, November 2002.

[4] X. Z. Gao, X. M. Gao, and S. J. Ovaska, "Modified Elman neural network model with application to dynamical systems identification," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1376–1381, October 1996.

[5] X. Wang, C. Zhang, and S. Zhang, "Modified Elman neural network and its application to network traffic prediction," in *Proceedings of the 2012 2nd IEEE International Conference on Cloud Computing and Intelligence Systems, IEEE CCIS 2012*, pp. 629–633, China, November 2012.

[6] W.-M. Lin and C.-M. Hong, "A new Elman neural network-based control algorithm for adjustable-pitch variable-speed wind-energy conversion systems," *IEEE Transactions on Power Electronics*, vol. 26, no. 2, pp. 473–481, 2011.

[7] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, December 1995.

[8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks (ICNN '95)*, vol. 4, pp. 1942–1948, Perth, Western Australia, November-December 1995.

[9] A. M. Sharaf and A. A. A. El-Gammal, "Particle Swarm Optimization PSO: A new search tool in power system and electro technology," *Studies in Computational Intelligence*, vol. 302, pp. 235–294, 2010.

[10] M. Saravanan, S. M. R. Slochanal, P. Venkatesh, and J. P. S. Abraham, "Application of particle swarm optimization technique for optimal location of FACTS devices considering cost of installation and system loadability," *Electric Power Systems Research*, vol. 77, no. 3-4, pp. 276–283, 2007.

[11] Y. Hou H, L. Lu J, and X. Xiong Y, "Enhanced particle swarm optimization algorithm and its application on economic dispatch of power systems[J]," in *Proceedings of the Csee*, vol. 24, pp. 95–100, 2004.

[12] T. Gao, J. L. Lan, and Y. F. Li, "Network traffic prediction with radial basis function neural network based on quantum adaptive particle swarm optimization," *Journal of Electronics Information Technology*, vol. 35, no. 9, pp. 2220–2226, 2013.

[13] J. Sun, W. Xu, and W. Fang, "Enhancing global search ability of quantum-behaved particle swarm optimization by maintaining diversity of the swarm," in *Rough sets and current trends in computing*, vol. 4259 of *Lecture Notes in Comput. Sci.*, pp. 736–745, Springer, Berlin, Germany, 2006.

[14] Y. Shi, G. D. Albada, J. Dongarra, and P. M. Sloot, *Computational Science – ICCS 2007*, vol. 4487, Springer, Berlin, Germany, 2007.

[15] 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.

[16] H.-W. D. Chiang, C.-C. Chen, and C.-N. Hsu, "Prediction of shrouded turbomachinery blade flutter by a complex mode analysis," *International Journal of Turbo and Jet Engines*, vol. 22, no. 2, pp. 89–101, 2005.

[17] S. H. Ling, H. H. C. Iu, F. H. F. Leung, and K. Y. Chan, "Improved hybrid particle swarm optimized wavelet neural network for modeling the development of fluid dispensing for electronic packaging," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 9, pp. 3447–3460, 2008.

[18] J. T Cheng, Z. M Duan, and Y. Xiong, "QAPSO-BP algorithm and its application in vibrationfault diagnosis for a hydroelectric generating unit," vol. 34, no. 23, pp. 177–182, 2015 (Chinese).

[19] F. Shi, X. C. Wang, and L. Yu, "MATLAB neural network 30 case analysis [M]," *Beijing University of Aeronautics and Astronautics Press*, pp. 183-184, 2010 (Chinese).

[20] S.-S. Zhong and Y. Li, "Condition monitoring of aeroengine based on wavelet process neural networks," *Hangkong Xuebao/Acta Aeronautica et Astronautica Sinica*, vol. 28, no. 1, pp. 68–71, 2007.

[21] A. J. Volponi, H. DePold, R. Ganguli, and C. Daguang, "The use of kalman filter and neural network methodologies in gas turbine performance diagnostics: a comparative study," *Journal of Engineering for Gas Turbines and Power*, vol. 125, no. 4, pp. 917–924, 2003.

[22] X. Fu, G. Ding, and S. Zhong, "Aeroengine turbine exhaust gas temperature prediction using support vector machines," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 5552, no. 2, pp. 235–241, 2009.