

Research Article

Aero Engine Fault Diagnosis Using an Optimized Extreme Learning Machine

Xinyi Yang,¹ Shan Pang,² Wei Shen,¹ Xuesen Lin,¹ Keyi Jiang,¹ and Yonghua Wang¹

¹Department of Aircraft Engineering, Naval Aeronautical and Astronautical University, Yantai 264001, China

²College of Information and Electrical Engineering, Ludong University, Yantai 264025, China

Correspondence should be addressed to Xinyi Yang; jimyang2008@163.com

Received 1 June 2015; Accepted 13 October 2015

Academic Editor: Roger L. Davis

Copyright © 2016 Xinyi Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new extreme learning machine optimized by quantum-behaved particle swarm optimization (QPSO) is developed in this paper. It uses QPSO to select optimal network parameters including the number of hidden layer neurons according to both the root mean square error on validation data set and the norm of output weights. The proposed Q-ELM was applied to real-world classification applications and a gas turbine fan engine diagnostic problem and was compared with two other optimized ELM methods and original ELM, SVM, and BP method. Results show that the proposed Q-ELM is a more reliable and suitable method than conventional neural network and other ELM methods for the defect diagnosis of the gas turbine engine.

1. Introduction

The gas turbine engine is a complex system and has been used in many fields. One of the most important applications of gas turbine engine is the propulsion system in aircraft. During its operation life, the gas turbine engine performance is affected by a lot of physical problems, including corrosion, erosion, fouling, and foreign object damage [1]. These may cause the engine performance deterioration and engine faults. Therefore, it is very important to develop engine diagnostics system to detect and isolate the engine faults for safe operation of an aircraft and reduced engine maintenance cost.

Engine fault diagnosis methods are mainly divided into two categories: model-based and data-driven techniques. Model-based techniques have advantages in terms of on-board implementation considerations. But they need an engine mathematical model and their reliability often decreases as the system nonlinear complexities and modeling uncertainties increase [2]. On the other hand, data-driven approaches do not need any system model and primarily rely on collected historical data from the engine sensors. They show great advantage over model-based techniques in many engine diagnostics applications. Among these data-driven approaches, the artificial neural network (ANN) [3, 4] and

the support vector machine (SVM) [5, 6] are two of the most commonly used techniques.

Applications of neural networks and SVM in engine fault diagnosis have been widely studied in the literature. Zedda and Singh [7] proposed a modular diagnostic system for a dual spool turbofan gas turbine using neural networks. Romessis et al. [8] applied a probabilistic neural network (PNN) to diagnose faults on turbofan engines. Volponi et al. [9] applied Kalman Filter and neural network methodologies to gas turbine performance diagnostics. Vanini et al. [2] developed fault detection and isolation (FDI) scheme for an aircraft jet engine. The proposed FDI system utilizes dynamic neural networks (DNNs) to simulate different operating model of the healthy engine or the faulty condition of the jet engine. Lee et al. [10] proposed a hybrid method of an artificial neural network combined with a support vector machine and have applied the method to the defect diagnostic of a SUAV gas turbine engine.

However, conventional ANN has some weak points: it needs many training data and the traditional learning algorithms are usually far slower than required. It may fall in the local minima instead of the global minima. In case of gas turbine engine diagnostics, however, the operating range is so wide. If the conventional ANN is applied to this

case, the classification performance may decrease because of the increasing nonlinearity of engine behavior in a wide operating range [11].

In recent years, a novel learning algorithm for single hidden layer neural networks called extreme learning machine (ELM) has been proposed and shows better performance on classification problem than many conventional ANN learning algorithms and SVM [12–14]. In ELM, the input weights and hidden biases are randomly generated, and the output weights are calculated by Moore-Penrose (MP) generalized inverse. ELM learns much faster with higher generalization performance than the traditional gradient-based learning algorithms such as back-propagation and Levenberg-Marquardt method. Also, ELM avoids many problems faced by traditional gradient-based learning algorithms such as stopping criteria, learning rate, and local minima problem.

Therefore ELM should be a promising method for gas turbine engine diagnostics. However ELM may require more hidden neurons than traditional gradient-based learning algorithms and lead to ill-conditioned problem because of the random selection of the input weights and hidden biases [15]. To address these problems, in this paper, we proposed an optimized ELM using quantum-behaved particle swarm optimization (Q-ELM) and applied it to the fault diagnostics of a gas turbine fan engine.

The rest of the paper is organized as follows. Section 2 gives a brief review of ELM. QPSO algorithm is overviewed in Section 3. Section 4 presents the proposed Q-ELM. Section 5 compares the Q-ELM with other methods on three real-world classification applications. In Section 6, Q-ELM is applied to gas turbine fan engine component fault diagnostics applications followed by the conclusions in Section 7.

2. Brief of Extreme Learning Machine

Extreme learning machine was proposed by Huang et al. [12]. For N arbitrary distinct samples $(\mathbf{x}_i, \mathbf{t}_i)$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{im}]^T \in \mathbf{R}^n$ and $\mathbf{t}_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in \mathbf{R}^m$, standard SLFN with K hidden neurons and activation function $g(x)$ can approximate these N samples with zero error which means that

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T}, \quad (1)$$

where $\mathbf{H} = \{h_{ij}\}$, ($i = 1, \dots, N$ and $j = 1, \dots, K$) is the hidden layer output matrix, $h_{ij} = g(\mathbf{w}_j \cdot \mathbf{x}_i + b_j)$ denotes the output of j th hidden neuron with respect to \mathbf{x}_i , and $\mathbf{w}_j = [w_{j1}, w_{j2}, \dots, w_{jm}]^T$ is the weight connecting j th hidden neuron and input neurons. b_j denotes the bias of j th hidden neuron. And $\mathbf{w}_j \cdot \mathbf{x}_i$ is the inner product of \mathbf{w}_j and \mathbf{x}_i . $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_k]^T$ is the matrix of output weights and $\boldsymbol{\beta}_j = [\beta_{j1}, \beta_{j2}, \dots, \beta_{jm}]^T$ ($j = 1, \dots, K$) is the weight vector connecting the j th hidden neuron and output neurons. And $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_N]^T$ is the matrix of desired output.

Therefore, the determination of the output weights is to find the least square (LS) solutions to the given linear system. The minimum norm LS solution to linear system (1) is

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^+ \mathbf{T}, \quad (2)$$

where \mathbf{H}^+ is the MP generalized inverse of matrix \mathbf{H} . The minimum norm LS solution is unique and has the smallest norm among all the LS solutions. ELM uses MP inverse method to obtain good generalization performance with dramatically increased learning speed.

3. Brief of Quantum-Behaved Particle Swarm Optimization

Recently some population based optimization algorithms have been applied to real-world optimization applications and show better performance than traditional optimization methods. Among them, genetic algorithm (GA) and particle swarm optimization (PSO) are two mostly used algorithms. GA was originally motivated by Darwin's natural evolution theory. It repeatedly modifies a population of individual solutions by three genetic operators: selection, crossover, and mutation operator. On the other hand, PSO was inspired by social behavior of bird flocking. However, unlike GA, PSO does not need any genetic operators and is simple in use compared with GA. The dynamics of population in PSO resembles the collective behavior of socially intelligent organisms. However, PSO has some problems such as premature or local convergence and is not a global optimization algorithm.

QPSO is a novel optimization algorithm inspired by the fundamental theory of particle swarm optimization and features of quantum mechanics [16]. The introduction of quantum mechanics helps to diversify the population and ameliorate convergence by maintaining more attractors. Thus, it improves the QPSO's performance and solves the premature or local convergence problem of PSO and shows better performance than PSO in many applications [17]. Therefore it is more suitable for ELM parameter optimization than GA and PSO.

In QPSO, the state of a particle \mathbf{y} is depicted by Schrodinger wave function $\psi(\mathbf{y}, t)$, instead of position and velocity. The dynamic behavior of the particle is widely divergent from classical PSO in that the exact values of position and velocity cannot be determined simultaneously. The probability of the particle's appearing in apposition can be calculated from probability density function $|\psi(\mathbf{y}, t)|^2$, the form of which depends on the potential field the particle lies in. Employing the Monte Carlo method, for the i th particle \mathbf{y}_i from the population, the particle moves according to the following iterative equation:

$$\begin{aligned} \mathbf{y}_{i,j}(t+1) &= \mathbf{P}_{i,j}(t) - \beta \cdot (\text{mBest}_j(t) - \mathbf{y}_{i,j}(t)) \\ &\quad \cdot \ln\left(\frac{1}{u_{i,j}(t)}\right) \quad \text{if } k \geq 0.5, \\ \mathbf{y}_{i,j}(t+1) &= \mathbf{P}_{i,j}(t) + \beta \cdot (\text{mBest}_j(t) - \mathbf{y}_{i,j}(t)) \\ &\quad \cdot \ln\left(\frac{1}{u_{i,j}(t)}\right) \quad \text{if } k < 0.5, \end{aligned} \quad (3)$$

where $y_{i,j}(t+1)$ is the position of the i th particle with respect to the j th dimension in iteration t . $\mathbf{P}_{i,j}$ is the local attractor of i th particle to the j th dimension and is defined as

$$\mathbf{P}_{i,j}(t) = \varphi_j(t) \cdot \text{pBest}_{i,j}(t) + (1 - \varphi_j(t)) \text{gBest}_j(t), \quad (4)$$

$$\text{mBest}_j(t) = \frac{1}{N_p} \sum_{i=1}^{N_p} \text{pBest}_{i,j}(t), \quad (5)$$

where N_p is the number of particles and pBest_i represents the best previous position of the i th particle. gBest is the global best position of the particle swarm. mBest is the mean best position defined as the mean of all the best positions of the population; k , u , and φ are random number distributed uniformly in $[0, 1]$, respectively. β is called contraction-expansion coefficient and is used to control the convergence speed of the algorithm.

4. Extreme Learning Machine Optimized by QPSO

Because the output weights in ELM are calculated using random input weights and hidden biases, there may exist a set of nonoptimal or even unnecessary input weights and hidden neurons. As a result, ELM may need more hidden neurons than conventional gradient-based learning algorithms and lead to an ill-conditioned hidden output matrix, which would cause worse generalization performance.

In this section, we proposed a new algorithm named Q-ELM to solve these problems. Unlike some other optimized ELM algorithms, our proposed algorithm optimizes not only the input weights and hidden biases using QPSO, but also the structure of the neural network (hidden layer neurons). The detailed steps of the proposed algorithm are as follows.

Step 1 (initializing). Firstly, we generate the population randomly. Each particle in the population is constituted by a set of input weights, hidden biases, and s -variables:

$$\mathbf{P}_i = [w_{11}, \dots, w_{NK}, b_1, \dots, b_K, s_1, \dots, s_K], \quad (6)$$

where s_i , $i = 1, \dots, h$, is a variable which defines the structure of the network. As illustrated in Figure 1, if $s_i = 0$, then the i th hidden neuron is not considered. Otherwise, if $s_i = 1$, the i th hidden neuron is retained and the sigmoid function is used as its activation function.

All components constituting a particle are randomly initialized within the range $[0, 1]$.

Step 2 (fitness evaluation). The corresponding output weights of each particle are computed according to (5). Then the fitness of each particle is evaluated by the root mean square error between the desired output and estimated output. To avoid the problem of overfitting, the fitness evaluation is performed on the validation dataset instead of the whole training dataset:

$$f(\mathbf{P}_i) = \sqrt{\frac{\sum_{j=1}^{N_v} \sum_{i=1}^K \beta_i g(\mathbf{w}_i \cdot \mathbf{w}_j + b_i) - \mathbf{t}_j}{N_v}}, \quad (7)$$

where N_v is the number of samples in the validation dataset.

Step 3 (updating pBest_i and gBest). With the fitness values of all particles in population, the best previous position for i th particle, pBest_i , and the global best position gBest of each particle are updated. As suggested in [15], neural network tends to have better generalization performance with the weights of smaller norm. Therefore, in this paper, the fitness value and the norm of output weights are considered together for updating pBest_i and gBest . The updating strategy is as follows:

$$\text{pBest}_i = \begin{cases} \mathbf{P}_i, & (f(\text{pBest}_i) - f(\mathbf{P}_i) > \eta f(\text{pBest}_i)) \text{ or } (|f(\text{pBest}_i) - f(\mathbf{P}_i)| < \eta f(\text{pBest}_i), \|\mathbf{w}_{\mathbf{P}_i}\| < \|\mathbf{w}_{\text{pBest}_i}\|) \\ \text{pBest}_i, & \text{else,} \end{cases} \quad (8)$$

$$\text{gBest} = \begin{cases} \mathbf{P}_i, & (f(\text{gBest}) - f(\mathbf{P}_i) > \eta f(\text{gBest})) \text{ or } (|f(\text{gBest}) - f(\mathbf{P}_i)| < \eta f(\text{gBest}), \|\mathbf{w}_{\mathbf{P}_i}\| < \|\mathbf{w}_{\text{gBest}}\|) \\ \text{gBest}, & \text{else,} \end{cases}$$

where $f(\mathbf{P}_i)$, $f(\text{pBest}_i)$, and $f(\text{gBest})$ are the fitness value of the i th particle's position, the best previous position of the i th particle, and the global best position of the swarm. $\mathbf{w}_{\mathbf{P}_i}$, $\mathbf{w}_{\text{pBest}_i}$, and $\mathbf{w}_{\text{gBest}}$ are the corresponding output weights of the position of the i th particle, the best previous position of the i th particle, and the global best position obtained by MP inverse. By this updating criterion, particles with smaller fitness values or smaller norms are more likely to be selected as pBest_i or gBest .

Step 4. Calculate each particle's local attractor \mathbf{P}_i and mean best position mBest according to (4) and (5).

Step 5. Update particle's new position according to (3).

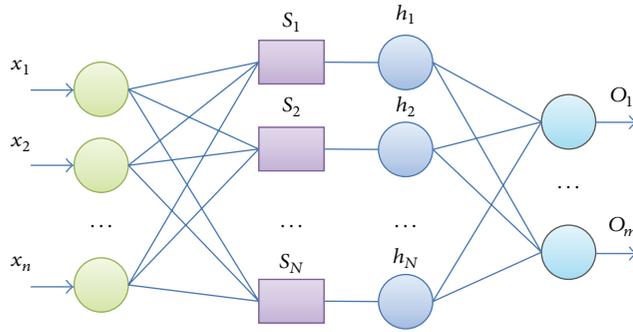
Finally, we repeat Step 2 to Step 5 until the maximum number of iterations is reached. Thus the network trained by ELM with the optimized input weights and hidden biases is obtained, and then the optimized network is applied to the benchmark problems.

TABLE 1: Specification of 3 classification problems.

Names	Attributes	Classes	Number of samples		
			Training data set	Validation set	Testing set
Satellite image	36	7	2661	1774	2000
Image segmentation	19	7	1000	524	786
Diabetes problem	8	2	346	230	192

TABLE 2: Mean training and testing accuracy of six methods on different classification problems.

Algorithms	Satellite image		Image segmentation		Diabetes problem	
	Training	Testing	Training	Testing	Training	Testing
Q-ELM	0.890	0.876	0.965	0.960	0.854	0.842
P-ELM	0.884	0.872	0.938	0.947	0.836	0.813
G-ELM	0.877	0.869	0.938	0.934	0.812	0.804
ELM	0.873	0.852	0.930	0.921	0.783	0.773
SVM	0.879	0.870	0.931	0.916	0.792	0.781
BP	0.856	0.849	0.919	0.892	0.785	0.776

FIGURE 1: Single hidden layer feedforward network with s -variable.

In the proposed algorithm, each particle represents one possible solution to the optimization problem and is a combination of components with different meaning and different range.

All components of a particle are firstly initialized into continuous values between 0 and 1. Therefore, before calculating corresponding output weights and fitness evaluation in Step 2, they need to be converted to their real value.

For the input weights and biases, they are given by

$$z_{ij} = (z_i^{\max} - z_i^{\min}) p_{ij} + z_i^{\min}, \quad (9)$$

where $z_i^{\max} = 1$ and $z_i^{\min} = -1$ are the upper and lower bound for input weights and hidden biases.

For s -parameters, they are given by

$$z_{ij} = \text{round}(p_{ij}), \quad (10)$$

where $\text{round}()$ is a function that rounds p_{ij} to the nearest integer (0 or 1, in this case). After the conversion of all components of a particle, the fitness of each particle can be then evaluated.

TABLE 3: Mean training time of six methods on different classification problems (in second).

Algorithms	Satellite image	Image segmentation	Diabetes problem
Q-ELM	178.56	45.36	19.23
P-ELM	198.33	39.08	20.69
G-ELM	274.07	54.26	22.12
ELM	0.034	0.027	0.011
SVM	9.54	3.61	1.43
BP	34.75	16.92	3.06

5. Evaluation on Some Classification Applications

In essence, engine diagnosis is a pattern classification problem. Therefore, in this section, we firstly apply the developed Q-ELM on some real-world classification applications and compare it with five existing algorithms. They are PSO optimized ELM (P-ELM) [18], genetic algorithm optimized (G-ELM) [18], standard ELM, BP, and SVM.

The performances of all algorithms are tested on three benchmark classification datasets which are listed in Table 1. The training dataset, validation dataset, and testing dataset are randomly generated at each trial of simulations according to the corresponding numbers in Table 1. The performances of these algorithms are listed in Tables 2 and 3.

For the three optimized ELMs, the population size is 100 and the maximum number of iterations is 50. The selection criteria for the P-ELMs and Q-ELM include the norm of output weights as (8), while the selection criterion for G-ELM considers only testing accuracy on validation dataset and does not include the norm of output weights as suggested in [19]. Instead, G-ELM incorporates Tikhonov's regularization in the least squares algorithm to improve the net generalization capability.

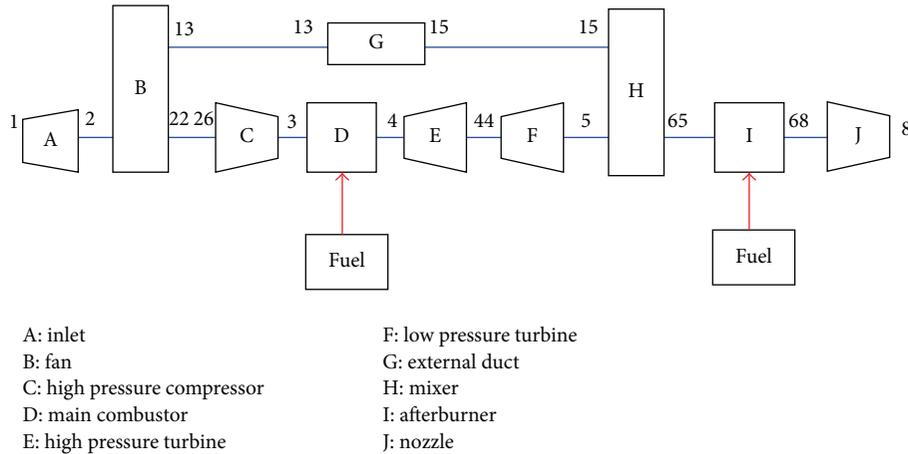


FIGURE 2: Schematic of studied turbine fan engine.

In G-ELM, the probability of crossover is 0.5 and the mutation probability is 10%. In Q-ELM and P-ELM, the inertial weight is set to decrease from 1.2 to 0.4 linearly with the iterations. In Q-ELM, the contraction-expansion coefficient β is set to decrease from 1.0 to 0.5 linearly with the iterations. ELM methods are set with different initial hidden neurons according to different applications.

There are many variants of BP algorithm; a faster BP algorithm called Levenberg-Marquardt algorithm is used in our simulations. And it has a very efficient implementation of Levenberg-Marquardt algorithm provided by MATLAB package. As SVM is binary classifier, here, the SVM algorithm has been expanded to “One versus One” Multiclass SVM to classify the multiple fault classes. The parameters for the SVM are $C = 10$ and $\gamma = 2$ [20]. The imposed noise level $N_l = 0.1$.

In order to account for the stochastic nature of these algorithms, all of the six methods are run 10 times separately for each classification problem and the results shown in Tables 2 and 3 are the mean performance values in 10 trials. All simulations have been made in MATLAB R2008a environment running on a PC with 2.5 GHz CPU with 2 cores and 2 GB RAM.

It can be concluded from Table 2 that, in general, the optimized ELM method obtained better classification results than ELM, SVM, and BP. Q-ELM outperforms all the other methods. It obtains the best mean testing and training accuracy on all these three classification problems. This suggests that Q-ELM is a good choice for engine fault diagnosis application.

Also it can be observed clearly that the training times of three optimized ELM methods are much more than the others. This mainly is because the optimized ELM methods need to repeatedly execute some steps of parameters optimization. And ELM costs the least training time among these methods.

6. Engine Diagnosis Applications

6.1. Engine Selection and Modeling. The developed Q-ELM was also applied to fault diagnostics of a gas turbine engine

TABLE 4: Description of two engine operating points.

Operating point	Fuel flow kg/s	Environment setting parameters	
		Velocity (Mach number)	Altitude (km)
A	1.142	0.8	5.8
B	1.155	1.2	6.9

and was compared with other methods. In this study, we focus on a two-shaft turbine fan engine with a mixer and an afterburner (for confidentiality reasons the engine type is omitted). This engine is composed of several components such as low pressure compressor (LPC) or fan, high pressure compressor (HPC), low pressure turbine (LPT), and high pressure turbine (HPT) and can be illustrated as shown in Figure 2.

The gas turbine engine is susceptible to a lot of physical problems and these problems may result in the component fault and reduce the component flow capacity and isentropic efficiency. These component faults can result in the deviations of some engine performance parameters such as pressures and temperatures across different engine components. It is a practical way to detect and isolate the default component using engine performance data.

But the performance data of real engine with component fault is very difficult to collect; thus the component fault is usually simulated by engine mathematical model as suggested in [10, 11].

We have already developed a performance model for this two-shaft turbine fan engine in MATLAB environment. In this study, we use the performance model to simulate the behavior of the engine with or without component faults.

The engine component faults can be simulated by isentropic efficiency deterioration of different engine components. By implanting corresponding component defects with certain magnitude of isentropic efficiency deterioration to the engine performance model, we can obtain simulated engine performance parameter data with component fault.

TABLE 5: Single and multiple fault cases.

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8
LPC	F	—	—	—	F	F	F	F
HPC	—	F	—	—	F	—	F	—
LPT	—	—	F	—	—	F	F	F
HPT	—	—	—	F	—	—	—	F

The engine operating point, which is primarily defined by fuel flow rate, has significant effect on the engine performance. Therefore, engine fault diagnostics should be conducted on a specified operating point. In this study, we study two different engine operation points. The fuel flow and environment setting parameters are listed in Table 4. The engine defect diagnostics was conducted on these operating points separately.

6.2. Generating Component Fault Dataset. In this study, different engine component fault cases were considered as the eight classes shown in Table 5. The first four classes represent four single fault cases. They are low pressure compressor (LPC) fault case, high pressure compressor (HPC) fault case, low pressure turbine (LPT) fault case, and high pressure turbine (HPT) fault case. Each class has only one component fault and is represented with an ‘‘F.’’ Class 5 and class 6 are dual fault cases, they are LPC + HPC fault case and LPC + LPT fault case. And the last two classes are triple fault cases. They are LPC + HPC + LPT fault case and LPC + LPT + HPT fault case.

For each single fault cases listed in Table 2, 50 instances were generated by randomly selecting corresponding component isentropic efficiency deterioration magnitude within the range 1%–5%. For dual fault cases, 100 instances were generated for each class by randomly setting the isentropic efficiency deterioration of two faulty components within the range 1%–5% simultaneously. For triple fault cases, each case generated 300 instances using the same method.

Thus we have 200 single fault data instances, 800 multiple fault instances, and one healthy state instance on each operating point condition. These instances are then divided into training dataset, validation dataset, and testing dataset.

In this study, for each operating point condition, 100 single fault case datasets (randomly select 25 instances for each single fault class), 400 multiple fault case datasets (randomly select 50 instances for each dual fault case and 150 instances for each triple fault case), and one healthy state instance were used as training dataset. 60 single fault case datasets (randomly select 15 instances for each single fault class) and 240 multiple fault case datasets (randomly select 30 instances for each dual fault case and 90 instances for each triple fault case) were used as testing dataset. And the left 200 instances were used as validation dataset.

The input parameters of the training, validation, and test dataset are the relative deviations of simulated engine performance parameters with component fault to the ‘‘healthy’’ engine parameters. And these parameters include low pressure rotor rotational speed n_1 , high pressure rotor rotational speed n_2 , total pressure and total temperature after LPC p_{22}^* ,

T_{22}^* , total pressure and total temperature after HPC p_3^* , T_3^* , total pressure and total temperature after HPT p_{44}^* , T_{44}^* , and total pressure and total temperature after LPT p_5^* , T_5^* . In this study, all the input parameters have been normalized into the range [0, 1].

In real engine applications, there inevitably exist sensor noises. Therefore, all input data are contaminated with measurement noise to simulate real engine sensory signals as the following equation:

$$X_n = X + N_l \cdot \sigma \cdot \text{rand}(1, n), \quad (11)$$

where X is clean input parameter, N_l denotes the imposed noise level, and σ is the standard deviation of dataset.

6.3. Engine Component Fault Diagnostics by 6 Methods. The proposed engine diagnostic method using Q-ELM is demonstrated with single and multiple fault cases and compared with P-ELM, G-ELM, ELM, BP, and SVM.

6.3.1. Parameter Settings. The parameter settings are the same as in Section 5 except that all ELM methods are set with 100 initial hidden neurons. All the six methods are run 10 times separately for each condition and the results shown in Tables 6, 7, and 8 are the mean performance values.

6.3.2. Comparisons of the Six Methods. The performances of the 6 methods were compared on both condition A and condition B. Tables 6 and 7 list the mean classification accuracies of the 6 methods on each component fault class. Table 8 lists the mean training time of each method.

It can be seen from Table 6 (condition A) that Q-ELM obtained the best results on C1, C4, C6, C7, and C8, while P-ELM performed the best on C2, C3, and C5. From Table 7, we can see that P-ELM performs the best on one single fault case (C4), and Q-ELM obtains the highest classification accuracy on all the left test cases.

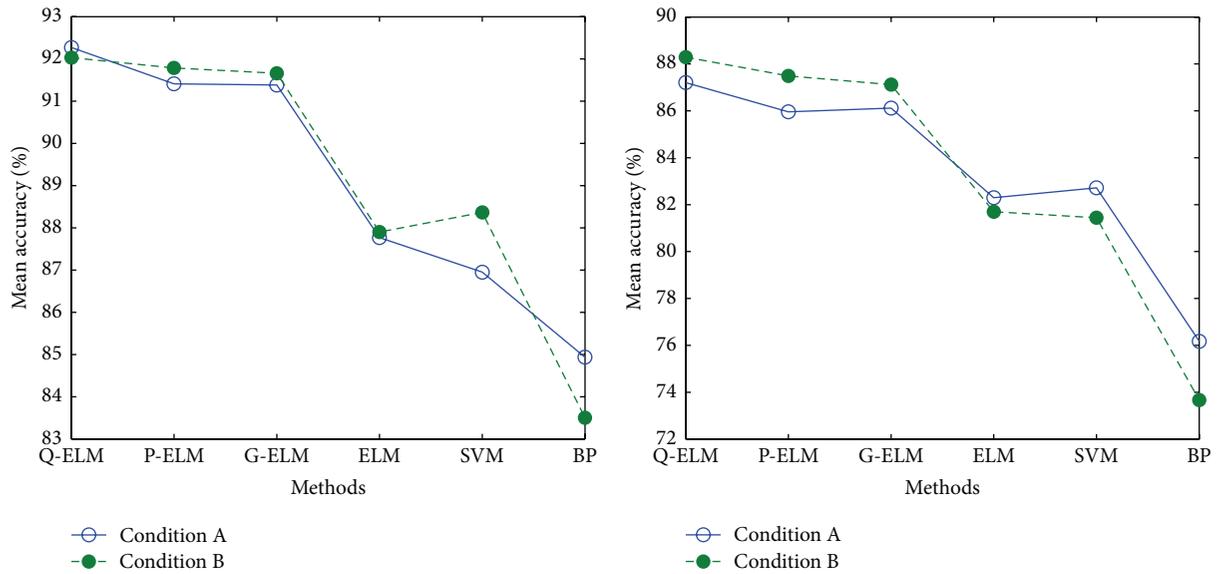
In general, the optimized ELMs obtained better classification results than ELM, SVM, and BP on both single fault and multiple fault cases. This conclusion can be also demonstrated in Figures 3(a) and 3(b), where the mean classification accuracies obtained by any optimized ELM methods are higher than that obtained by the other three methods. It can be also observed that the classification performance of ELM is on par with that of SVM and is better than that of BP. Due to the nonlinear nature of gas turbine engine, the multiple fault cases are more difficult to diagnose than single fault cases. It can be seen from Tables 3 and 4 that the mean classification accuracy of multiple fault diagnostics is lower than that of single fault diagnostics cases.

TABLE 6: Mean classification accuracy of operating point A condition.

Method	Fault classes							
	C1	C2	C3	C4	C5	C6	C7	C8
Q-ELM	93.33	93.55	91.75	90.45	90.05	88.56	85.26	84.95
P-ELM	90.48	93.61	92.32	89.23	90.44	87.73	83.05	82.60
G-ELM	92.35	93.04	90.17	89.36	88.24	88.03	84.69	83.50
ELM	89.09	87.56	87.52	86.90	85.27	84.42	81.16	78.33
SVM	89.03	88.67	84.96	85.15	85.68	85.02	80.54	79.63
BP	87.29	86.45	82.37	83.63	80.57	78.69	72.20	73.24

TABLE 7: Mean classification accuracy of operating point B condition.

Method	Fault classes							
	C1	C2	C3	C4	C5	C6	C7	C8
Q-ELM	93.45	92.07	91.53	91.07	91.45	90.19	86.26	85.26
P-ELM	92.57	91.58	91.51	91.48	89.63	89.70	86.18	84.44
G-ELM	93.31	91.74	90.53	91.06	91.23	88.65	85.75	82.83
ELM	89.21	88.33	86.97	87.09	85.95	83.55	80.24	77.06
SVM	90.66	89.14	87.14	86.52	85.68	82.12	81.42	76.54
BP	86.25	84.51	82.55	80.69	76.05	74.63	73.04	70.95



(a) Mean classification accuracies on single fault cases of the 6 methods (b) Mean classification accuracies on multiple fault cases of the 6 methods

FIGURE 3: Mean classification accuracies of the 6 methods of different conditions.

TABLE 8: Mean training time on training data.

	Q-ELM	G-ELM	P-ELM	ELM	SVM	BP
Condition A	19.54	23.08	20.77	0.0113	0.147	2.96
Condition B	20.23	23.54	21.83	0.0108	0.179	3.13

Notice that the two mean accuracy curves on both Figures 3(a) and 3(b) are very close to each other; we can conclude that engine operating point has no obvious effect on classification accuracies of all methods.

The training times of three optimized ELM methods are much more than the others. Much of training time of the optimized ELM is spent on evaluating all the individuals iteratively.

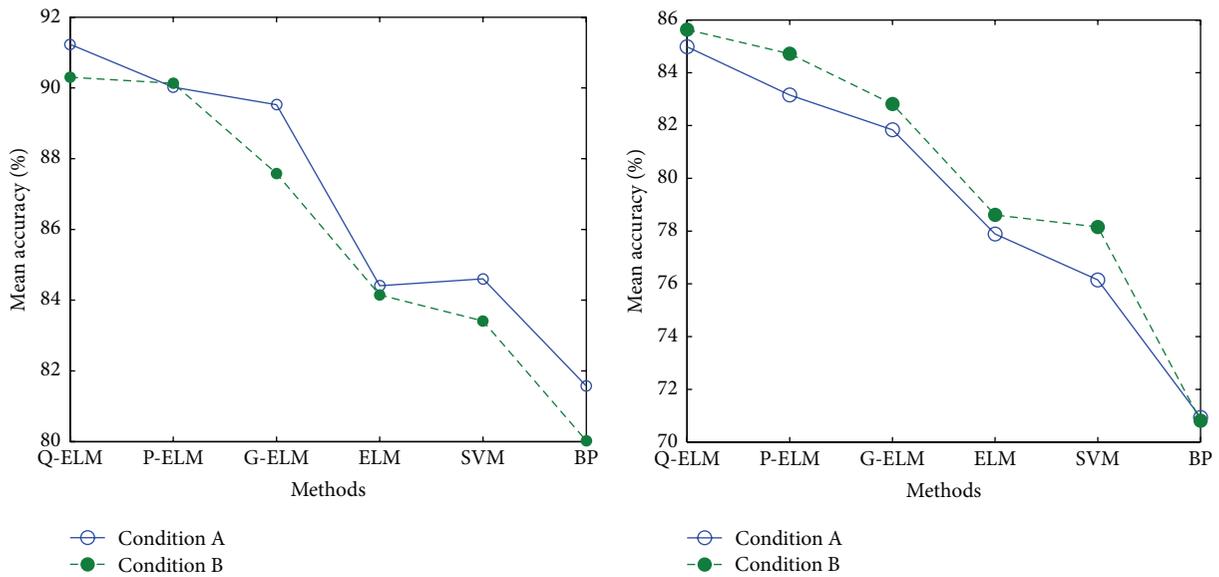
6.3.3. Comparisons with Fewer Input Parameters. In Section 6.3.2 we train ELM and other methods using dataset with 10 input parameters. But, in real applications, the gas turbine engine may be equipped with only a few numbers of sensors. Thus we have fewer input parameters. In this section we reduce the input parameters from 10 to 6; they are $n_1, n_2,$

TABLE 9: Mean classification accuracy of operating point A condition.

Method	Fault classes							
	C1	C2	C3	C4	C5	C6	C7	C8
Q-ELM	92.0857	91.8108	91.0178	90.0239	88.8211	86.3334	83.0490	81.7370
P-ELM	90.0024	90.8502	91.1142	88.1223	86.7068	85.2295	80.5144	80.1667
G-ELM	89.0763	91.0038	87.7888	88.2232	85.4190	83.9324	79.6553	78.3395
ELM	85.8684	84.8761	83.4979	82.3035	81.6548	78.2046	76.1726	74.7309
SVM	83.9047	85.8948	84.4922	83.2766	81.1377	74.6795	74.5938	74.1796
BP	83.2354	82.2824	80.9588	79.8117	72.3411	71.9088	70.9969	68.4934

TABLE 10: Mean classification accuracy of operating point B condition.

Method	Fault classes							
	C1	C2	C3	C4	C5	C6	C7	C8
Q-ELM	91.6970	90.3428	89.8130	89.3616	87.7345	86.4981	84.6418	83.6606
P-ELM	91.5233	90.1717	89.6428	89.1923	86.5645	86.3305	82.4815	83.5021
G-ELM	88.9153	87.6171	87.1090	86.6763	85.0338	84.8484	81.1512	80.2104
ELM	84.2722	83.9392	84.0175	82.9732	80.9403	80.5231	77.0267	75.9607
SVM	84.1035	82.8468	83.3550	82.9361	80.2821	81.1347	76.5557	74.6450
BP	80.6848	80.4343	79.9449	79.0281	74.8724	71.0306	69.1692	68.1630



(a) Mean classification accuracies on single fault cases of the 6 methods (b) Mean classification accuracies on multiple fault cases of the 6 methods

FIGURE 4: Mean classification accuracies of the 6 methods of different conditions.

TABLE 11: Mean training time on training data.

	Q-ELM	G-ELM	P-ELM	ELM	SVM	BP
Condition A	17.40	21.26	18.75	0.0094	0.126	2.35
Condition B	18.79	20.55	19.06	0.0077	0.160	2.47

p_{22}^* , p_3^* , T_{44}^* , and p_5^* . We trained all the methods with the same training dataset with only 6 input parameters and the results are listed in Tables 9–11.

Compared with Tables 6 and 7, we can see that the number of input parameters has great impact on diagnostics accuracy. The results in Tables 9 and 10 are generally lower than results in Tables 6 and 7.

The optimized ELM methods still show better performance than the others, this conclusion can be demonstrated in Figures 4(a) and 4(b). Q-ELM obtained the highest accuracies in all cases except C3 in Table 6 and it attained the best results in all cases in Table 10.

The good results obtained by our method indicate that the selection criteria which include both the fitness value in

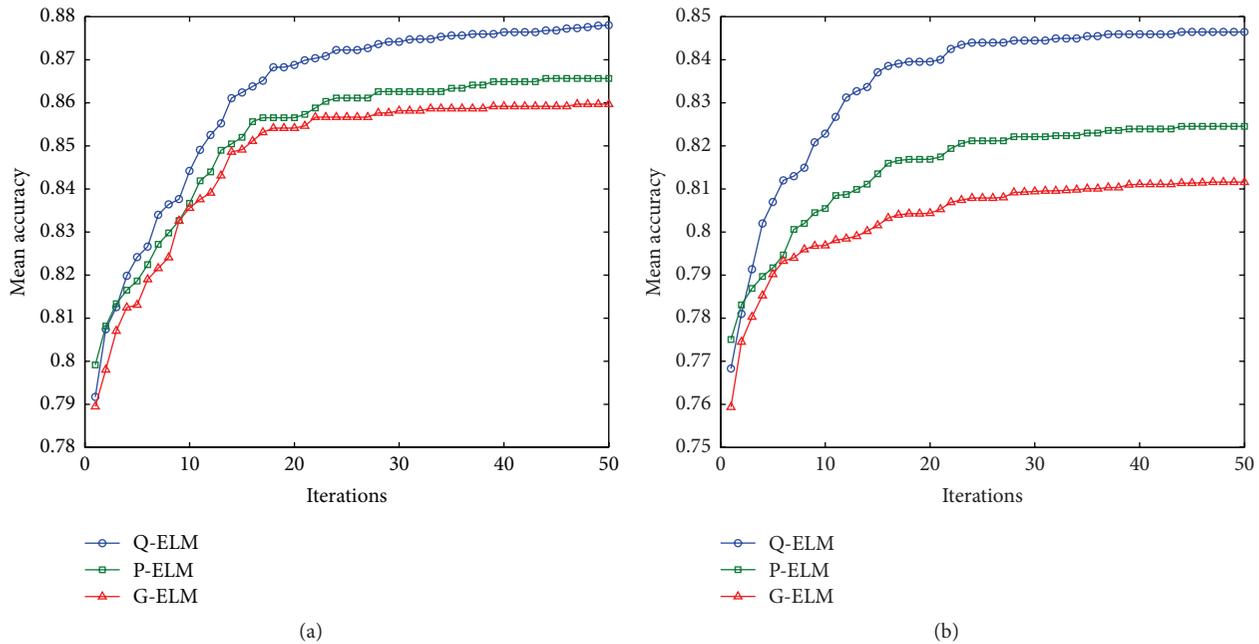


FIGURE 5: Mean evolution of the RMSE of the three methods: (a) fault class C5 and (b) fault class C7.

validation dataset and the norm of output weights help the algorithms to obtain better generalization performance.

In order to evaluate the proposed method in depth, the mean evolution of the accuracy on validation dataset of 10 trials by three optimized ELM methods on fault class C5 and C7 cases in condition B is plotted in Figure 5.

It can be observed from Figure 5 that Q-ELM has much better convergence performance than the other two methods and obtains the best mean accuracy after 50 iterations; P-ELM is better than G-ELM. In fact, Q-ELM can achieve the same accuracy level as G-ELM within only half of the total iterations for these two cases.

The main reason for high classification rate by our method is mainly because the quantum mechanics helps QPSO to search more effectively in search space, thus outperforming P-ELMs and G-ELM in converging to a better result.

7. Conclusions

In this paper, a new hybrid learning approach for SLFN named Q-ELM was proposed. The proposed algorithm optimizes both the neural network parameters (input weights and hidden biases) and hidden layer structure using QPSO. And the output weights are calculated by Moore-Penrose generalized inverse, like in the original ELM. In the optimizing of network parameters, not only the RMSE on validation dataset but also the norm of the output weights is considered to be included in the selection criteria.

To validate the performance of the proposed Q-ELM, we applied it to some real-world classification applications and a gas turbine fan engine fault diagnostics and compare it with some state-of-the-art methods. Results show that our method obtains the highest classification accuracy in most test cases and show great advantage than the other optimized

ELM methods, SVM and BP. This advantage becomes more prominent when the number of input parameters in training dataset is reduced, which suggests that our method is a more suitable tool for real engine fault diagnostics application.

Conflict of Interests

The authors declare no conflict of interests regarding the publication of this paper.

References

- [1] X. Yang, W. Shen, S. Pang, B. Li, K. Jiang, and Y. Wang, "A novel gas turbine engine health status estimation method using quantum-behaved particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2014, Article ID 302514, 11 pages, 2014.
- [2] Z. N. S. Vanini, K. Khorasani, and N. Meskin, "Fault detection and isolation of a dual spool gas turbine engine using dynamic neural networks and multiple model approach," *Information Sciences*, vol. 259, pp. 234–251, 2014.
- [3] T. Brotherton and T. Johnson, "Anomaly detection for advanced military aircraft using neural networks," in *Proceedings of the IEEE Aerospace Conference*, vol. 6, pp. 3113–3123, IEEE, Big Sky, Mont, USA, March 2001.
- [4] S. Ogaji, Y. G. Li, S. Sampath, and R. Singh, "Gas path fault diagnosis of a turbofan engine from transient data using artificial neural networks," in *Proceedings of the 2003 ASME Turbine and Aeroengine Congress*, ASME Paper No. GT2003-38423, Atlanta, Ga, USA, June 2003.
- [5] L. C. Jaw, "Recent advancements in aircraft engine health management (EHM) technologies and recommendations for the next step," in *Proceedings of the 50th ASME International Gas Turbine & Aeroengine Technical Congress*, Reno, Nev, USA, June 2005.

- [6] S. Osowski, K. Siwek, and T. Markiewicz, "MLP and SVM networks—a comparative study," in *Proceedings of the 6th Nordic Signal Processing Symposium (NORSIG '04)*, pp. 37–40, Espoo, Finland, June 2004.
- [7] M. Zedda and R. Singh, "Fault diagnosis of a turbofan engine using neural-networks: a quantitative approach," in *Proceedings of the 34th AIAA, ASME, SAE, ASEE Joint Propulsion Conference and Exhibit*, AIAA 98-3602, Cleveland, Ohio, USA, July 1998.
- [8] C. Romessis, A. Stamatis, and K. Mathioudakis, "A parametric investigation of the diagnostic ability of probabilistic neural networks on turbofan engines," in *Proceedings of the ASME Turbo Expo 2001: Power for Land, Sea, and Air*, Paper no. 2001-GT-0011, New Orleans, La, USA, June 2001.
- [9] 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.
- [10] S.-M. Lee, T.-S. Roh, and D.-W. Choi, "Defect diagnostics of SUAV gas turbine engine using hybrid SVM-artificial neural network method," *Journal of Mechanical Science and Technology*, vol. 23, no. 1, pp. 559–568, 2009.
- [11] S.-M. Lee, W.-J. Choi, T.-S. Roh, and D.-W. Choi, "A study on separate learning algorithm using support vector machine for defect diagnostics of gas turbine engine," *Journal of Mechanical Science and Technology*, vol. 22, no. 12, pp. 2489–2497, 2008.
- [12] 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.
- [13] G.-B. Huang, Z. Bai, L. L. C. Kasun, and C. M. Vong, "Local receptive fields based extreme learning machine," *IEEE Computational Intelligence Magazine*, vol. 10, no. 2, pp. 18–29, 2015.
- [14] G.-B. Huang, "What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John Von Neumann's puzzle," *Cognitive Computation*, vol. 7, no. 3, pp. 263–278, 2015.
- [15] 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.
- [16] J. Sun, C.-H. Lai, W.-B. Xu, Y. Ding, and Z. Chai, "A modified quantum-behaved particle swarm optimization," in *Proceedings of the 7th International Conference on Computational Science (ICCS '07)*, pp. 294–301, Beijing, China, May 2007.
- [17] 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.
- [18] T. Matias, F. Souza, R. Araújo, and C. H. Antunes, "Learning of a single-hidden layer feedforward neural network using an optimized extreme learning machine," *Neurocomputing*, vol. 129, pp. 428–436, 2014.
- [19] F. Han, H.-F. Yao, and Q.-H. Ling, "An improved evolutionary extreme learning machine based on particle swarm optimization," *Neurocomputing*, vol. 116, pp. 87–93, 2013.
- [20] D.-H. Seo, T.-S. Roh, and D.-W. Choi, "Defect diagnostics of gas turbine engine using hybrid SVM-ANN with module system in off-design condition," *Journal of Mechanical Science and Technology*, vol. 23, no. 3, pp. 677–685, 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

