ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 3 - 2026

Rolling Bearings Fault Diagnosis Using an Optimized Extreme Learning Machine by CBQGA

Hong Zhang[1,*], *Yiqi Zhou*[1], *Sheng Ma*[2] *and Zhengqing Zhu*[1]

[1] Sino-European Institute of Aviation Engineering, Civil Aviation University of China, Tianjin 300300, China
[2] College of Aeronautical Engineering, Civil Aviation University of China, Tianjin, 300300

SUMMARY: *We propose an Extreme Learning Machine (ELM) optimized by a quantum genetic algorithm formulated on Bloch-sphere Coordinates with Chaotic Mutation (CBQGA), hereafter referred to as CBQGA-ELM. CBQGA selects the ELM hyperparameters—including the number of hidden neurons, input weights, and hidden biases—by minimizing the Root Mean Square Error (RMSE) on a validation set. The resulting model is utilized to rolling bearing fault diagnosis. First, CBQGA-ELM is benchmarked against Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Quantum Genetic Algorithm (QGA), and Bloch-quantum Genetic Algorithm (BQGA) optimizer using datasets from the UCI Machine Learning Repository. Simulation results show that CBQGA delivers superior optimization performance relative to these alternatives. Second, laboratory experiments are conducted in which vibration signals are collected for four bearing conditions: healthy, inner race fault, outer race fault, and ball fault. Time-domain features are extracted from the signals and supplied to the diagnostic models. The results demonstrate that CBQGA-ELM achieves higher diagnostic accuracy and reliability than the competing methods, indicating its suitability for rolling bearing defect diagnosis.*

KEYWORDS: *Bloch-quantum genetic algorithm; Chaotic mutation; Rolling bearing; Fault diagnosis; Extreme learning machine*

1 Introduction

The aero-engine is a highly complex system and one of the most critical components of an aircraft. Throughout its service life, aero-engine performance is significantly influenced by harsh operating demands, including strong vibrations and elevated temperatures [1]. The rotor system, in particular, is vulnerable to failure under these conditions, leading to performance degradation and potential faults. Consequently, the developing and reliable diagnostic systems for fault detection and isolation is essential to ensure safe aircraft operation and reduce maintenance costs [2]. Rolling bearings, as key elements of the engine rotor system, play a decisive role in overall engine reliability and efficiency [3]. Even the bearings of other same types may exhibit markedly different service lives, which has drawn considerable research attention both domestically and internationally.

With the rapid advancement of Artificial Intelligence (AI), such technologies are increasingly applied to aero-engine rolling bearing fault diagnosis. Numerous studies have investigated different approaches for processing vibration signals and integrating them with

*h_zhang@cauc.edu.cn
https://doi.org/10.65102/is20261112
Published: 10 June, 2026

data-driven fault diagnosis techniques [4], including support vector machines [5–7], multiple kernel learning [8], artificial neural networks [9,10], Elman networks [11], and radial basis function (RBF) networks [12], achieving promising results.

Feedforward Neural Networks (FFNNs) are widely utilized in diagnostic applications due to their ability to approximate complex nonlinear relationships directly from input data. However, their training process is typically slow, which has limited their practical application [13]. To restrain this defect, the Extreme Learning Machine (ELM) was proposed as a fast-learning alternative [14]. ELM is successfully utilized in data compression, feature extraction, clustering, diagnostics et al., as it alleviates several limitations of FFNNs, including sensitivity to learning rate, termination criteria, and susceptibility to local minima [15]. Nonetheless, ELM often demands multiple hidden neurons, and its reliance on random initialization of input weights and hidden biases can result in ill-conditioning [16]. To overcome these defects, various optimization techniques are applied. For example, Xu et al. employed whale optimization algorithms, and Yang et al. utilized quantum-behaved particle swarm optimization for aero-engine fault diagnosis [17]. Additionally, Lu et al. introduced sensor failure efficiency into ELM to enhance its diagnostic and reconstruction capabilities [18].

Benefiting from on the fundamental conclusion, this paper put forwards a novel diagnostic approach— the Chaotic Bloch Quantum Genetic Algorithm optimized ELM (CBQGA-ELM)—to address the limitations of ELM in bearing fault diagnosis [19]. The Quantum Genetic Algorithm (QGA) integrates quantum computing principles into conventional Genetic Algorithm (GA), exploiting quantum parallelism to improve data-processing capacity. However, QGA owns drawbacks such as frequent binary decoding and reliance on predefined rotation tables [20]. To restrain these limitations, the Bloch Quantum Genetic Algorithm (BQGA) was introduced, employed phase-reversal mutations within a defined search region [21]. To further elevate chromosome diversity, chaotic mutation is incorporated, resulting in the Chaotic Bloch Quantum Genetic Algorithm (CBQGA) [22].

In CBQGA-ELM, the input-to-hidden weights and hidden biases are optimized using CBQGA, while the output weights are computed analytically via the Moore–Penrose pseudoinverse. This approach is then utilized to diagnose rolling bearing faults in aero-engines.

This paper owns the following structures: Section 2 provides an overview of ELM, while Section 3 introduces the fundamentals of CBQGA. Section 4 describes the CBQGA-ELM optimization method, incorporating qubit Bloch-coordinate representation and chaotic mutation. Section 5 details the vibration experiments on rolling bearings. Section 6 outlines the architecture of CBQGA-ELM, and Section 7 presents its application to bearing fault diagnosis along with experimental results. Finally, Section 8 concludes the whole paper.

2 Brief of Extreme Learning Machine

In ELM, the input–hidden weights and hidden biases are assigned randomly, and the output weights are computed analytically. With any bounded, piecewise-continuous nonlinear activation, a single-hidden-layer network trained in this manner retains universal approximation capability [23].

The architecture of ELM is shown in Figure 1, which can be mathematically represented by (1) in [24]:

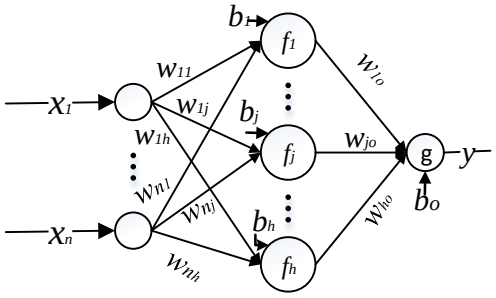

Figure 1: Network architecture of ELM.

$$y = g\left(b_o + \sum_{j=1}^{h} w_{jo} v_j \right)$$
$$v_j = f_j\left(b_j + \sum_{i=1}^{n} w_{ij} x_i \right) \tag{1}$$

n and h are the input variables' number and the hidden layer neurons' number, respectively; v_j means the output of hidden-layer neurons j; x_i stands for the input variables, i=1,2,3....n; w_{ij} is the connecting weight between the hidden layer's input variable i and the neurons j; w_{jo} stands for the connecting weight between j of the hidden layer and output neurons; b_j is the hidden layer neurons' bias, j=1,2,3....h; b_o means the output layer neurons' bias; $f_j(\cdot)$ is the hidden layer neurons' activation function; $g(\cdot)$ stands for the output layer neurons' activation function.

Considering that N samples are available, when $b_o = 0$ and activation function is linear, (1) can be enunciated as:

$$y = \left(w_o{}^T v \right)^T \tag{2}$$

$y = [y(1)\ y(2) \cdots y(N)]^T$ stands for an outputting vector; $w_o = [w_{1o}\ w_{2o} \cdots w_{ho}]^T$ is an outputting weight's vector; v means an outputting matrix of hidden layer neurons; inputting weights and bias matrix w is randomly assigned. v and w is given by (3):

$$v = \begin{bmatrix} v_1(1) & v_1(2) & \cdots & v_1(N) \\ v_2(1) & v_2(2) & \cdots & v_2(N) \\ \vdots & \vdots & & \vdots \\ v_h(1) & v_h(2) & \cdots & v_h(N) \end{bmatrix} \quad w = \begin{bmatrix} b_1 & b_2 & \cdots & b_h \\ w_{11} & w_{12} & \cdots & w_{1h} \\ \vdots & \vdots & & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nh} \end{bmatrix} \tag{3}$$

The output weights vector w_o can be estimated by:

$$\hat{w}_o = v^{\dagger} y_d \tag{4}$$

$v^{\dagger}$ stands for the output matrix v 's Moore-Penrose (MP) generalized inverse; $y_d = [y_d(1)\ y_d(2) \cdots y_d(N)]^T$ is the desired output. When $v \in R^{N \times h}$, $N \geq h$ and $rank(v) = h$, the Moore-Penrose (MP) generalized inverse $v^{\dagger}$ of v can be calculated by:

$$v^{\dagger} = (v^T v)^{-1} v^T \tag{5}$$

Substituting (5) into (4), the estimation of can be obtained using a Least-Squares (LS) solution:

$$\hat{w}_o = (v^T v)^{-1} v^T y_d \tag{6}$$

The minimum norm LS solution is one-of-a-kind, which owns the minimal one. ELM utilizes MP inverse means to attain good generalization ability. Therefore, ELM is more effective than BP or SVM to diagnose rolling bearings faults.

Some hidden layer activation function used in this paper. As follows:

Sigmoid () function:

$$f(a,b,x) = \frac{1}{1+e^{(-(ax+b))}}$$

Sin () function:

$$f(a,b,x) = \sin(ax+b)$$

RBF () function:

$$f(a,b,x) = e^{(-b\|x-a\|^2)}$$

Hardlim () function:

$$f(a,b,x) = \begin{cases} 1 & if\ ax-b \geq 0 \\ 0 & otherwise \end{cases}$$

3 The principle of CBQGA

Intelligent optimization algorithms, like GA and Particle Swarm Optimization (PSO), have been utilized to optimize ELM. However, their optimization performance is often limited due to inherent shortcomings, including susceptibility to local optima [25, 26]. To address these limitations, a novel approach known as the quantum-behaved genetic algorithm (QGA) was proposed in, which integrates quantum computation into conventional GA [27]. By leveraging quantum parallelism, QGA significantly accelerates information processing. Unlike traditional GA, where binary chromosomes are employed, QGA represents chromosomes with qubits and updates populations through quantum gate operations. This design enhances global search capability, as qubit-based chromosomes can encode multiple quantum states simultaneously, allowing small populations of quantum individuals to replace large traditional populations.

Nevertheless, conventional QGA still suffers from several issues. First, frequent binary decoding increases computational overhead. Second, the rotation angle of the quantum revolving gate is typically determined by fixed lookup tables, limiting adaptability. Third, quantum states are described on a flat unit circle using only a single variable, which weakens their quantum characteristics. To overcome these drawbacks, BQGA was introduced in. In BQGA, quantum chromosome mutation is realized through phase reversal within the search region, while chaotic mutation perturbs the phase with limited amplitude to enhance

chromosome diversity. Building on this foundation, CBQGA is proposed by incorporating chaotic mutation into BQGA for further performance improvement.

3.1 Three means for quantum chromosome

In quantum computation, the smallest unit of information is expressed by qubit. The state of a qubit can be expressed by:

$$|\phi\rangle = \cos(\theta/2)|0\rangle + e^{i\varphi}\sin(\theta/2)|1\rangle \tag{7}$$

where θ and φ is phase angle of the point P on a Bloch sphere in Figure 2.

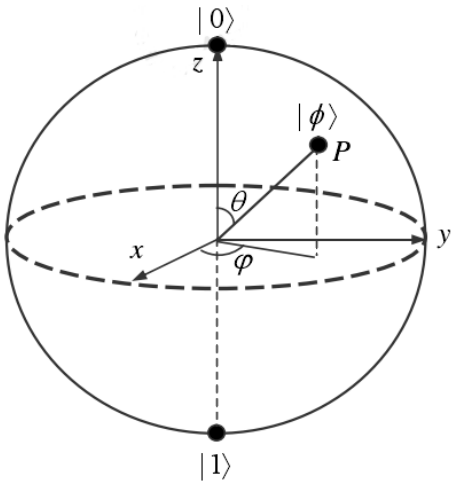

Figure 2: Bloch sphere representation of a qubit.

According to Figure 2, each qubit owns a corresponding point on this sphere. When θ and φ of a point were given, the point can be described as $|\phi\rangle = [\cos\varphi\sin\theta\ \sin\varphi\sin\theta\ \cos\theta]^T$. In CBQGA, qubits' Bloch coordinates can be deemed as genes on chromosome. P_i stands for populations' chromosome i, the populations of encoding method can be obtained:

$$P_i = \begin{vmatrix} \cos\varphi_{i1}\sin\theta_{i1} & \cdots & \cos\varphi_{in}\sin\theta_{in} \\ \sin\varphi_{i1}\sin\theta_{i1} & \cdots & \sin\varphi_{in}\sin\theta_{in} \\ \cos\theta_{i1} & \cdots & \cos\theta_{in} \end{vmatrix} \tag{8}$$

where $\varphi_{ij} = 2\pi \times rand$, $\theta_{ij} = \pi \times rand$, *rand* means a random number in (0,1); $i = 1,2,\cdots,sizepop$, *sizepop* represents population size, $j = 1,2,\cdots,numsum$, *numsum* represents the number of qubits (it is equivalent to optimize variables' number). In CBQGA, each quantum chromosomes contains three paratactic gene chains, and each gene chain stands for an optimized outcome. These gene chains can be obtained:

$$\begin{cases} p_{ix} = (\cos\varphi_{i1}\sin\theta_{i1},\cdots,\cos\varphi_{in}\sin\theta_{in}) \\ p_{iy} = (\sin\varphi_{i1}\sin\theta_{i1},\cdots,\cos\varphi_{in}\sin\theta_{in}) \\ p_{iz} = (\cos\theta_{i1},\cos\theta_{i2},\cdots,\cos\theta_{in}) \end{cases} \tag{9}$$

3.2 Solution space transform

Each chromosome in colony contains 3*n* Bloch coordinates of *n* qubits that can be transformed from unit space to solution space. Assume that range value of the *j*th variable for the optimization problem is $\mathrm{var}_j \in (a_j, b_j)$. If the *j*th qubit on chromosome P_i is $[x_{ij}, y_{ij}, z_{ij}]^T$, then the corresponding variables in solution space are, respectively, computed as follows:

$$\begin{cases} X_{ix}^{j} = [b_j(1+x_{ij}) + a_j(1-x_{ij})]/2 \\ X_{iy}^{j} = [b_j(1+y_{ij}) + a_j(1-y_{ij})]/2 \\ X_{iz}^{j} = [b_j(1+z_{ij}) + a_j(1-z_{ij})]/2 \end{cases} \tag{10}$$

where $i = 1, 2, \cdots, sizepop$, *sizepop* represents population size, $j = 1, 2, \cdots, numsum$, *numsum* represents the number of qubits (it is equivalent to optimize the number of variables). So, each chromosome has the optimized problem's corresponding three approximate solutions.

3.3 Quantum chromosome update

To ensure the optimal solution in space can be searched and the phase of the quantum chromosome needs to rotate the phases of qubits. Using quantum rotation gate *U* to update an individual qubit, *U* can be expressed:

$$U = \begin{bmatrix} \cos\Delta\varphi\cos\Delta\theta & -\sin\Delta\varphi\cos\Delta\theta & \sin\Delta\theta\cos(\Delta\varphi+\varphi) \\ \sin\Delta\varphi\cos\Delta\theta & \cos\Delta\varphi\cos\Delta\theta & \sin\Delta\theta\sin(\Delta\varphi+\varphi) \\ -\sin\Delta\varphi & -\tan(\varphi/2)\sin\Delta\theta & \cos\Delta\theta \end{bmatrix} \tag{11}$$

where $\Delta\theta$ and $\Delta\varphi$ represent the step of rotation of phase. When *U* premultiplication a certain qubit, it can be obtained:

$$U \cdot \begin{bmatrix} \cos\varphi\sin\theta \\ \sin\varphi\sin\theta \\ \cos\theta \end{bmatrix} = \begin{bmatrix} \cos(\varphi+\Delta\varphi)\sin(\theta+\Delta\theta) \\ \cos(\varphi+\Delta\varphi)\sin(\theta+\Delta\theta) \\ \cos(\theta+\Delta\theta) \end{bmatrix} \tag{12}$$

Clearly, *U* causes the phase rotation of $\Delta\theta$ and $\Delta\varphi$, but the convergence speed and effect are affected by the rotational angle's direction. Therefore, this direction is determined:

$$A = \begin{vmatrix} x_0 & x_1 \\ y_0 & y_1 \end{vmatrix} \tag{13}$$

$$B = z_0 - z_1 \tag{14}$$

The direction of $\Delta\varphi$ is determined by such rules as follows. If $A \neq 0$, then the direction of $\Delta\varphi$ is -sign(A); otherwise, this direction is sign(rand). Similarly, this outcome of $\Delta\theta$ can be obtained. If $B \neq 0$, then the direction of $\Delta\theta$ is -sign(B); otherwise, this direction is is sign(rand).

3.4 Chaotic mutation

To increase the population variation's diversity, one chaotic sequence is introduced to execute an perturb the phase with limited amplitude for the quantum chromosomes in the current algebra. Due to the effect of the different chaotic sequence is equivalent, so the chaotic sequence can be obtained by the Logist function in this paper. The disturbance amplitude is auto-adjusted according fitness value. The rules are as follows:

$$\lambda_i = e^{\frac{b_j - \min f(p_i)}{b_j}} \tag{15}$$

then, the perturbed phase θ and φ can be expressed as:

$$\begin{cases} \Delta\theta_{i,j} = \lambda_i \cdot \theta_{i,j-1} \cdot (1 - \theta_{i,j-1}) \\ \Delta\varphi_{i,j} = \lambda_i \cdot \varphi_{i,j-1} \cdot (1 - \varphi_{i,j-1}) \end{cases} \tag{16}$$

The chaotic mutation operation can avoid premature convergence and increase diversity of colony.

4 Brief of CBQGA-ELM

4.1 Introducing of CBQGA-ELM

Because ELM computes the outputting weights by using randomly assigned inputting weights and hidden biases, many of those parameters can be suboptimal—or even superfluous—along with redundant hidden neurons. Consequently, ELM often needs a larger hidden layer than gradient-based methods and can suffer from ill-conditioning.

To mitigate these limitations, a diagnostic framework termed CBQGA-ELM is put forwards, in which a Chaotic–Bloch Quantum Genetic Algorithm optimizes the ELM parameters. The computational workflow (illustrated in Figure. 3) comprises two coordinated components: the left panel implements the ELM-based diagnosis, while the right panel executes the CBQGA optimization. The procedure unfolds as follows.

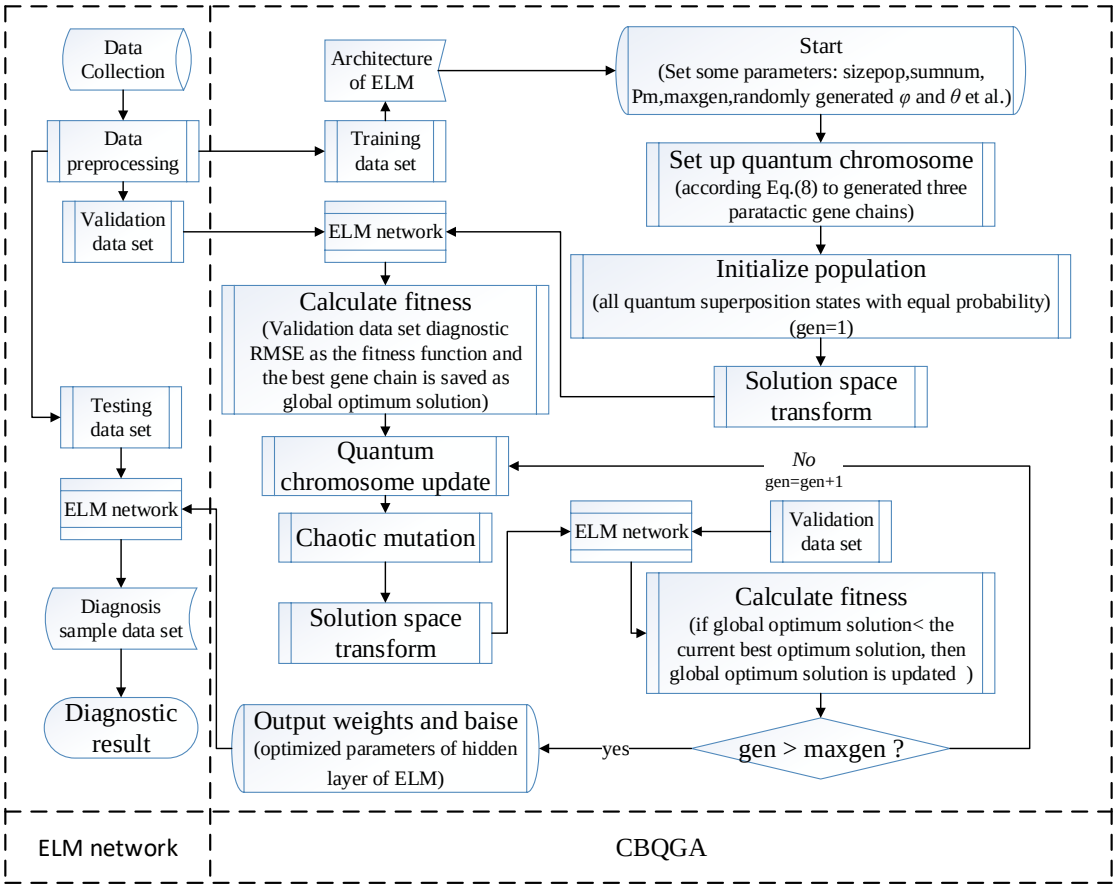

Figure 3: CBQGA-ELM flow chart.

Step 1: Setting parameters. Set the mutation probability *Pm*, size of population *sizepop*, total number of variables *sumnum*, step length of rotational phase angles, maximum optimization steps *maxgen* and so on.

Step 2: Initialization. Set *gen*=0. According Eq. (8) to randomly generate an initial population.

Step 3: Transform solution space. For each chromosome, the three gene chains—each representing a candidate solution—are mapped into the problem's solution space using Eq. (10). This mapping produces the set of candidate solutions for the optimization objective.

Step 4: Calculating fitness. Calculating the fitness of 3 *sizepop* approximate solutions according Eq. (17), the current optimum solutions and the current optimum chromosome are obtained. And saving these current optimum solutions as global optimum solutions.

$$fitness(\cdot) = RMSE = \sqrt{\frac{\sum_{i=1}^{n}(o_i - y_i)^2}{n}} \tag{17}$$

where, n is means the validation data set number; y_i stands for the diagnosis methods' expected output; o_i is the actual output of the diagnosis methods

Step 5: Updating chromosome. *gen*=*gen*+1. According Eq. (11) and Eq. (16) to update and mutate population, the new population is obtained.

Step 6: Transform solution space. Three new gene chains are transformed to the optimization problem's solution space according to Eq. (10), obtaining the goal set.

Step 7: Calculating fitness. Calculating this new approximate outcome, a current optimum solution and its chromosome can be attained.

Step 8: If the global optimum solution < the current optimum solution, then the global optimum solution is replaced by the current optimum solution. Else goes to next step.

Step 9: Loop to process 5, unless this means convergences or *gen>maxgen*.

4.2 Evaluating Classifying Applications

Engine fault diagnosis can fundamentally be formulated as a pattern classification problem. To this end, CBQGA-ELM on classifying tasks is done, comparing its performance with alternative approaches, including BQGA-ELM, QGA-ELM, PSO-ELM, GA-ELM, and the baseline ELM (collectively referred to as "diagnostic methods" for brevity).

Experiments are conducted on two benchmark datasets from the UCI Machine Learning Repository (Table 1). As described in Table 2, each dataset is randomly divided into mutually exclusive training, validation, and test subsets. Comparative performance results are reported in Tables 3 and 4, while Fig. 4 illustrates the RMSE convergence behavior.

Table 1: Test data set

Data set	Number of instance	Number of Attributes	Date Donated	Associated Tasks	Number of categories
Wine	178	13	1991/7/1	Classification	3
Abalone	4177	8	1995/12/1	Classification	3

Table 2: Size of Samples

Data set	Training data set	Samples number	Testing Data
Wine	100	20	58
Abalone	2500	600	1077

Parameters of these algorithm setting instructions, the same parameters of the genetic series algorithms are listed: the population size is 100, the maximum iteration is 100, the crossover's probability is 0.5, the mutation's probability is 0.05. In PSO-ELM, the population particles number is 20, the maximum number of iterations is 100, the maximum particle velocity is 0.1, the minimum velocity is -0.1, the particle range is [-0.5 0.5]. In CBQGA-ELM, the probability of mutation is 0.05, the step length of rotational phase angles are both 0.01π. Sigmoid() function, Sin() function Hardlim() function and RBF() function are used in ELM and optimized ELM network. The hidden layer neurons of ELM and optimized ELM are given in the Table 3 and Table 4.

To capture algorithmic randomness, each method is executed independently 30 times on every classifying task, and Tables 3–4 report the mean performance across these trials. All trials were conducted in MATLAB R2014a on a desktop equipped with a 2.5 GHz quad-core CPU and 4 GB of RAM.

Table 3: Diagnostic results of Wine

Activation function	Algorithm	Testing date Accuracy (%)	Root mean square error	Number of hidden neurons	Computational Time (in second)
Sigmoid()	CBQGA-ELM	93.45%	0.2183	37	70.831643
	BQGA-ELM	81.03%	0.3257	30	50.981838
	QGA-ELM	70%	0.5504	25	20.857564
	PSO-ELM	85.83%	0.4448	25	50.343029
	GA-ELM	81.67%	0.3813	30	30.844246
	ELM	76.45%		45	0.078942
Sin()	CBQGA-ELM	87.24%	1.0586	52	80.903587
	BQGA-ELM	83.79%	1.3386	40	60.757153
	QGA-ELM	74.17%	1.5989	55	30.834089
	PSO-ELM	81.67%	1.4206	37	40.620598
	GA-ELM	78.33%	1.4452	20	10.287021
	ELM	80.32%		34	0.068942
Hardlim()	CBQGA-ELM	96.90%	0.3585	39	60.56237
	BQGA-ELM	92.069	0.3502	30	50.694911
	QGA-ELM	50%	0.7504	35	30.660038
	PSO-ELM	94.67%	0.4476	39	40.488367
	GA-ELM	90.67%	0.4861	25	10.285705
	ELM	87.43%		32	0.067642
RBF()	CBQGA-ELM	80.34%	1.3558	40	60.547073
	BQGA-ELM	87.24%	1.2379	50	70.261577
	QGA-ELM	65%	1.6882	25	20.922956
	PSO-ELM	86.67%	1.3459	40	40.279434
	GA-ELM	70.34%	1.5092	65	10.415685
	ELM	68.23%		23	0.058965

Table 4: Diagnostic results of Abolone

Activation function	Algorithm	Testing date Accuracy (%)	Root mean square error	Number of hidden neurons	Computational Time (in minute)
Sigmoid()	CBQGA-ELM	96.45%	0.118345	307	30.345
	BQGA-ELM	97.03%	0.097234	408	36.64
	QGA-ELM	80%	0.35614	603	48.23
	PSO-ELM	95.83%	0.100448	279	20.681
	GA-ELM	81.67%	0.63813	400	31.84
	ELM	86.45%		507	8.78
Hardlim()	CBQGA-ELM	98.90%	0.003585	390	45.312
	BQGA-ELM	92.07%	0.3502	502	50.692
	QGA-ELM	63%	1.7504	360	30.67
	PSO-ELM	94.67%	0.4456	490	36.72
	GA-ELM	90.02%	0.8861	259	20.283
	ELM	82.43%		349	5.063
RBF()	CBQGA-ELM	87.04%	1.4058	500	56.236
	BQGA-ELM	91.24%	0.92379	603	60.264
	QGA-ELM	75%	1.7282	300	20.924
	PSO-ELM	76.67%	1.7459	408	40.279
	GA-ELM	73.34%	1.54392	651	31.41
	ELM	58.23%		200	3.059

As observed from Table 3 and 4, the modified ELM methods gained better classification results than ELM. CBQGA-ELM outperforms all the other methods with different activation function in small sample set of Wine. In the lager sample set of Abolone, BQGA-ELM and CBQGA-ELM obtained better classification results than others under Sigmoid () and Hardlim () function respectively. The hidden-layer sizes differ between the baseline ELM and its optimized variants. Moreover, the modified ELM methods require substantially longer running times than standard ELM, primarily because they repeatedly parameter-optimizing steps.

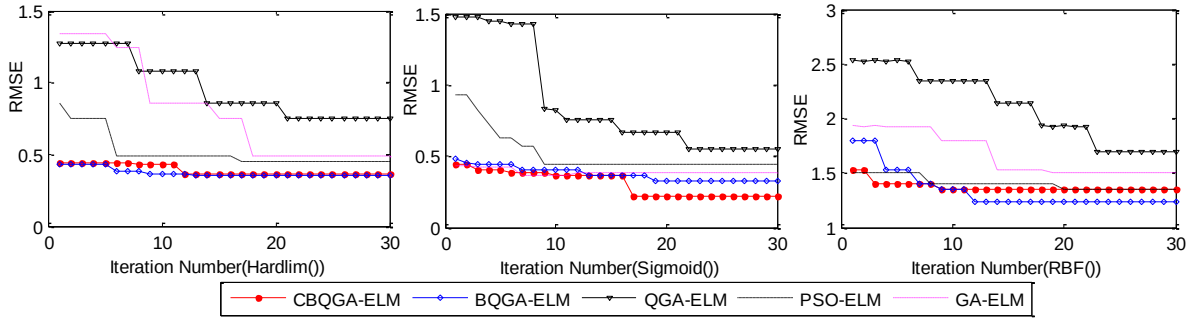

Figure 4: Average testing RMSE of different algorithms.

Figure 4 reveals the modified algorithms' converging curves in the Wine data set. The converging speed and values of CBQGA-ELM and BQGA-ELM is better than others under Hardlim(), Sigmoid() and RBF() function.

Through the above analysis, we need to choose suitable activation function and set suitable neurons of hidden layer when we use ELM network to address some classification problem, at same time, optimized ELM may obtain better results than standard ELM. Therefore, CBQGA-ELM is suitable for engine fault diagnosing.

5 Rolling Bearing Fault Experiment

To diagnose the rolling bearing fault, firstly, needs to collect the faulted rolling bearing's vibration signals from the Lab. Secondly, the vibration signals is processed by time-domain analysis to abstract fault characteristic data sets. Finally, these fault characteristic data sets are input to optimized ELM methods to train and diagnoses. In this section, Rolling Bearing Fault Experiment and collecting data are introduced. Some set and output rules of network and so on are described in the section 6. Diagnosis results and analyses are given in section 7.

5.1 Brief of rolling bearing fault experiment

It is difficult to collect direct vibration signals of aero-bearings, due to its location is not adapt to monitor and there is so much noise signal for the aero-engine. Therefore, rolling bearing fault data sets are obtained by experimental simulation, and its experiment platform is shown in Figure 5.

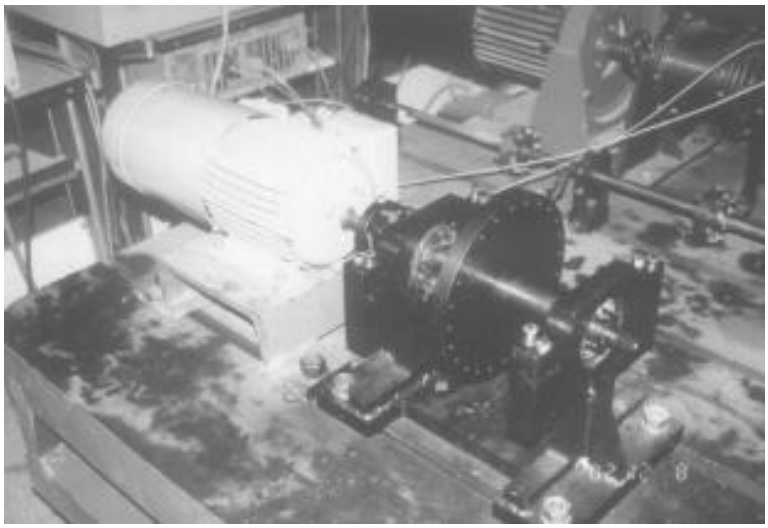

Figure 5: The test rig of the rolling element bearing.

Some devices are used in the experiment includes: type 307 rolling bearing, acceleration sensor, micro sensor, charge, direct current machine, signal conditioners, computer and so on. The bearing housing can be disassembled for easy replacement of bearings of different types of failure. Reflective sheeting is attached on the coupling as the phase start mark. The rotational speed and key phase signals can be obtained by the micro sensor. The bearing vibration signal is measured by the acceleration sensor placed in the axial direction, radial vertical direction and horizontal direction. These signals are amplified in the charge amplifier before it is sent to the CAMD-6100 signal conditioner for processing.

In the experiment, the rotation speed of the rotor is 988r/min, there are 1024 points per cycle were collected by the sensor, and 16 cycles were collected for each experiment. The geometric parameters of type 307 rolling bearing as shown in Table 5. The type of experimental failures include: normal rolling bearing, inner ring failure, outer ring fault and ball fault. The size data of these fault types is shown in Table 6, the pictures of failure type are shown in Figure 6, 7, and 8.

Ball diameter Bearing inner diameter Contact Angle Fault width description.

Table 5: The geometrical parameter of type 307 rolling bearing

Name of parameters	Size (mm)
Number of ball	7
Ball diameter	14.5
Bearing inner diameter	35
Bearing outer diameter	80
Contact angle ( )	0

Table 6: Different types of simulated faults on the rolling element bearing

The type of failures	Fault width (mm)	Description
inner ring fault	0.25	scratch
outer ring fault	0.25	scratch
ball fault	0.5	scratch

Note: In Table 6, the fault depth is all 1mm, and the axial length is 4mm.

Figure 6: The single scratch fault on the inner ring.

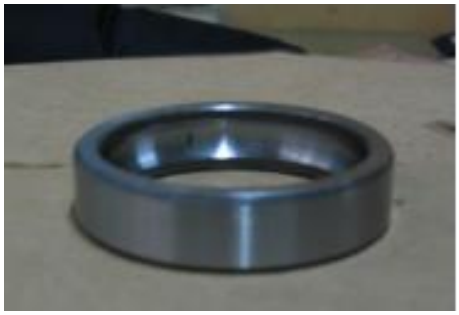

Figure 7: The single scratch fault on the outer ring.

Figure 8: The single erosion fault on a ball.

5.2 Vibration signal processing

The peak-to-peak value and Kurtosis coefficients that were obtained by the time-domain analysis are non-dimensional parameter, and time-domain analysis does not distort or lose for the vibration signal. Therefore, the time-domain is chosen to process vibration signal in this paper. Firstly, using the time-domain analysis analyzes vibration signal to obtain some parameters. Secondly, using factor analysis in SAS software to choose part factor that cumulative variance contribution rate reaches 90% is selected according to the variance of the variance of the gravel chart factor, namely: peak-to-peak value, Kurtosis coefficients, mean, root-mean-square, variance, skewness. Finally, these factors are used as the fault diagnosing input of type 307 rolling bearings. Due to the large amount of data, 100 samples of each type of fault are randomly selected as diagnostic cases. The partial data samples obtained after time domain analysis are shown in Table 7. (Note: the four types of fault are numbered as 1, 2, 3, and 4 respectively)

Table 7: Fault characteristic quantity of type 307 rolling bearing

No.	peak-to-peak value	mean	root-mean-square	variance	skewness	Kurtosis	Type.
1	0.2272	0.0612	-0.1794	0.0423	0.0103	0.0037	Normal (No:1)
2	0.2272	0.0686	-0.0919	-0.0933	0.0107	0.0047	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
99	0.22781	0.065264	-0.15715	-0.025115	0.012739	0.0042573	
100	0.21154	0.063019	-0.22925	0.16828	0.01282	0.0039694	
1	2.0301	0.17405	-0.027341	32.167	0.0030455	0.030279	Inner ring fault (No:2)
2	1.6555	0.15498	-0.22999	21.74	0.0032096	0.024007	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
99	1.1726	0.1513	-0.1608	11.6026	0.0043	0.0229	
100	1.4002	0.1662	-0.1985	16.1121	0.0043	0.0276	
1	1.1612	0.18131	0.086093	4.3799	0.0043866	0.032859	Ball fault (No:3)
2	0.55731	0.1062	0.092951	2.1073	0.0046804	0.011274	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
99	0.716	0.1468	0.0741	2.0106	0.0047	0.0215	
100	0.648	0.1188	0.1622	2.253	0.0044	0.0141	
1	0.3259	0.093	0.0421	-0.118	0.0034	0.0087	Outer ring fault (No:4)
2	0.3376	0.0948	-0.0004	-0.1776	0.0033	0.009	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
99	0.36466	0.093521	0.054205	-0.11369	0.0032307	0.0087419	
100	0.39584	0.094459	0.079657	0.043554	0.0030497	0.0089181	

5.3 Distribution of sample set

The data sample sets are divided into three parts, including the training, validating and testing data sets. The three parts of the data are randomly selected from initial data sets and non-overlap with each other. The proportion of these training and testing data sets (hereinafter referred to as “sample proportion”) influences the diagnosis accuracy, therefore, different diagnosis results under different the sample proportion are presented in this paper. When the sample proportion is 2:1, the number of data sets for each sample sets is shown in the Table 8.

Table 8: Size of Samples

Algorithms	Training data set	Samples number of validating set	Testing data set
CBQGA-ELM	200	100	100
BQGA-ELM			
QGA-ELM			
PSO-ELM			
GA-ELM			

6 Rolling bearing fault diagnosis Application

6.1 Data standardization

To meet the physical meaning and adapt to the training, these data require normalization.

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

$x_{\max}$ and $x_{\min}$ are a factor's maximum and minimum values (x_i is the one factor's ith value), x is the value obtained after normalization.

6.2 Treatment of output results of diagnosis methods

The diagnosis methods adopt single output mode, so, its output is not the *No.* of a certain class. In the paper, treatment of output results is shown in Table 9.

Table 9: Deal with way of ELM output

Output range of diagnosis methods	Classification
0~1	1
1~2	2
2~3	3
3~4	4
4~	*NaN*

6.3 Choice of activation function and number of hidden neurons

In section 4.2, the classifying accuracy of diagnosis methods can be influenced by activation function and the number of hidden neurons, therefore, selecting suitable activation function and the hidden neurons number is crucial. Due to there is no general method to obtain these two variables, experiment is conducted. The optimal hidden neurons number and activating function are selected as the one which results in the best testing accuracy. Figure 9 presents the outcomes for CBQGA-ELM, BQGA-ELM, QGA-ELM, PSO-ELM, GA-ELM and ELM with different activation function (including, Hardlim (), RBF(), Sigmoid(), Sin()). (Note: parameter setting of in this section and section 7 are the same as is section 4.2)

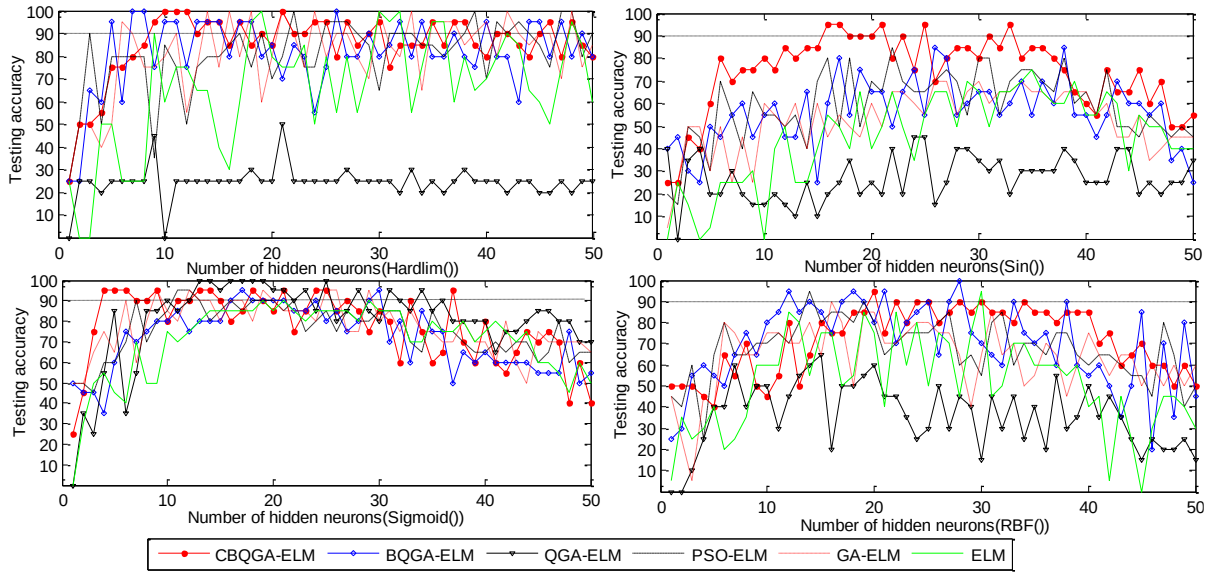

Figure 9: Activation function and hidden neurons number.

In that Figure 9, when the activation function are Hardlim() and Sigmoid() function, the diagnosis accuracy for testing data set of most diagnosis methods can reach to 90%. Simply,

the rolling bearings fault diagnosis accuracy under Hardlim() and Sigmoid() function are discussed. From Figure 9, when the number o hidden neurons of the diagnosis methods with Hardlim() and Sigmoid() function are within the ranges [10,45] and [10,30], respectively, these diagnosis methods can be reach to better accuracy. Therefore, the diagnosis methods' optimal hidden neurons number can be attained.

6.4 Evaluating the results

The mean diagnosing accuracy was attained:

$$Accuarcy_{mean} = \frac{1}{T}\sum_{t=1}^{T}(\frac{y(t)}{m}\times 100\%) \tag{19}$$

where, $Accuracy_{mean}$ stands for the mean accuracy; T stands for the trials number; $y(t)$ stands for the right diagnosing number of the *t*th trial; the samples' number need to diagnose is m.

7 Analysis of diagnosis results

In that section, the performance of all diagnosis methods are evaluated and compared. For the rolling bearing diagnosis problem, the results are averaged over 30 trails. The sample proportion, activation function, accuracy of training and testing data, validation RMSE, computational time, and the hidden neurons' number for each diagnosis means are shown.

7.1 Diagnosis accuracy

Table 10 to 12 summarizes the results for rolling bearing fault diagnosis aiming at the sample proportion, activation function, accuracy of training and testing data, validation RMSE, computational time, and the hidden neurons' number for each diagnosis means.

Table 10: The diagnostic accuracy between different diagnostic methods

Sample proportion	Activation function	Diagnosis method	Accuracy of training data	Accuracy of testing data	Validation RMSE	Computational time (in second)	Number of hidden neurons
3:1	Hardlim()	CBQGA-ELM	91.67%	95%	0.1408	19.180403	10
		BQGA-ELM	89.33%	84%	0.1417	18.816641	
		QGA-ELM	25%	26%	1.118	7.055107	
		PSO-ELM	96.67%	80%	0.3199	17.659255	
		GA-ELM	71.67%	78%	0.4262	6.992264	
		ELM	50.33%	52		0.144676	
	Sigmoid()	CBQGA-ELM	92.33%	89%	0.3143	19.763271	
		BQGA-ELM	93.33%	90%	0.14283	18.690582	
		QGA-ELM	88%	88%	0.2261	7.467908	
		PSO-ELM	97.67%	88%	0.3336	19.873848	
		GA-ELM	94%	78%	0.3284	6.13311	
		ELM	88.67%	70%		0.103345	
2:1	Hardlim()	CBQGA-ELM	96.60%	81.85%	0.0643	19.500197	10
		BQGA-ELM	95.85%	82.96%	0.0254	17.764383	
		QGA-ELM	19.62%	20.74%	1.1405	7.406023	
		PSO-ELM	99.62%	78.52%	0.3366	14.210962	
		GA-ELM	74.34%	63.70%	0.4258	5.772352	
		ELM	53.96%	52.59%		0.096465	
	Sigmoid()	CBQGA-ELM	92.45%	79.63%	0.2594	18.847828	
		BQGA-ELM	99.62%	87.41%	0.2232	18.443862	
		QGA-ELM	90.57%	76.30%	0.2547	7.258816	
		PSO-ELM	93.96%	73.33%	0.3698	13.525043	
		GA-ELM	95.85%	72.59%	0.3228	6.742839	
		ELM	90.94%	74.07%		0.096202	
1:1	Hardlim()	CBQGA-ELM	90.50%	82%	0.0642	17.961369	10
		BQGA-ELM	98%	66%	0.15794	18.10183	
		QGA-ELM	25%	25%	1.1346	7.377017	
		PSO-ELM	100%	72%	0.1543	18.725374	
		GA-ELM	73.50%	69%	0.2743	6.373321	
		ELM	51.50%	44.50%		0.099968	
	Sigmoid()	CBQGA-ELM	91%	76%	0.1996	17.842478	
		BQGA-ELM	97.50%	70%	0.1909	18.621302	
		QGA-ELM	90.50%	74%	0.1794	7.100111	
		PSO-ELM	98%	75.50%	0.2996	17.468785	
		GA-ELM	98%	71%	0.2723	6.273346	
		ELM	88%	51.50%		0.104257	

Table 11: The diagnostic accuracy between different diagnostic methods

Sample proportion	Activation function	Diagnosis method	Accuracy of training data	Accuracy of testing data	Validation RMSE	Computational time (in second)	Number of hidden neurons
3:1	Hardlim()	CBQGA-ELM	100%	96%	0.20780	22.665124	20
		BQGA-ELM	98%	82%	0.0203	23.774173	
		QGA-ELM	25%	24%	1.11800	8.724658	
		PSO-ELM	96%	81%	0.23090	21.386323	
		GA-ELM	81.33%	82%	0.25160	6.642623	
		ELM	73%	58%		0.119871	
	Sigmoid()	CBQGA-ELM	96%	89%	0.26380	23.635186	
		BQGA-ELM	100%	80%	0.08740	23.145163	
		QGA-ELM	96.67%	93%	0.27500	9.625318	
		PSO-ELM	100%	86%	0.2682	21.800866	
		GA-ELM	100%	88%	0.2563	7.202627	
		ELM	100%	90%		0.109653	
2:1	Hardlim()	CBQGA-ELM	100%	90%	0.00192	22.046882	20
		BQGA-ELM	98.11%	86.67%	0.01156	21.303557	
		QGA-ELM	24.53%	25.19%	1.14050	8.952353	
		PSO-ELM	98.49%	72.59%	0.24610	16.165975	
		GA-ELM	98.49%	72.59%	0.06860	6.143635	
		ELM	78.11%	73.33%		0.109333	
	Sigmoid()	CBQGA-ELM	99.25%	84.44%	0.32880	23.46158	
		BQGA-ELM	100%	82.22%	0.26730	21.313084	
		QGA-ELM	98.11%	83.70%	0.38150	8.6568	
		PSO-ELM	100%	75.56%	0.48290	16.02311	
		GA-ELM	99.62%	74.81%	0.36230	6.347916	
		ELM	99.62%	73.33%		0.118701	
1:1	Hardlim()	CBQGA-ELM	93%	76.50%	0.01356	20.972993	20
		BQGA-ELM	95%	73%	0.00924	20.211401	
		QGA-ELM	25%	24.50%	1.13530	9.819712	
		PSO-ELM	100%	89%	0.10750	19.223133	
		GA-ELM	95%	80.50%	0.11480	6.420021	
		ELM	85.50%	75%		0.110201	
	Sigmoid()	CBQGA-ELM	99.50%	71.50%	0.29320	22.478721	
		BQGA-ELM	100%	73%	0.19070	20.982051	
		QGA-ELM	100%	68%	0.27270	8.871717	
		PSO-ELM	100%	62.50%	0.32830	19.135029	
		GA-ELM	100%	71.50%	0.32480	6.741091	
		ELM	100%	67.50%		0.121815	

Table 12: The diagnostic accuracy between different diagnostic methods

Sample proportion	Activation function	Diagnosis method	Accuracy of training data	Accuracy of testing data	Validation RMSE	Computational time (in second)	Number of hidden neurons
3:1	Hardlim()	CBQGA-ELM	100%	95%	0.0257	24.348405	20
		BQGA-ELM	97%	77%	0.02100	26.373072	
		QGA-ELM	25%	27%	1.118	9.412456	
		PSO-ELM	100%	87%	0.1436	22.637448	
		GA-ELM	96.33%	82%	0.0768	6.933855	
		ELM	77.67%	70%		0.133175	
	Sigmoid()	CBQGA-ELM	100%	92%	0.4242	24.379992	
		BQGA-ELM	100%	84%	0.4097	24.981468	
		QGA-ELM	100%	84%	0.5157	9.539425	
		PSO-ELM	100%	79%	0.8406	21.203723	
		GA-ELM	100%	79%	0.7566	6.989307	
		ELM	100%	81%		0.110916	
2:1	Hardlim()	CBQGA-ELM	97.36%	91.85%	0.0021809	22.18445	20
		BQGA-ELM	100%	87.04%	0.090277	22.064523	
		QGA-ELM	24.53%	23.70%	1.1405	9.363341	
		PSO-ELM	98.87%	77.04%	0.1932	16.470409	
		GA-ELM	90.57%	71.11%	0.2393	5.768233	
		ELM	72.08%	70.37%		0.113304	
	Sigmoid()	CBQGA-ELM	100%	80%	0.24627	22.993316	
		BQGA-ELM	100%	68.89%	0.6691	22.766713	
		QGA-ELM	99.62%	77.04%	0.9836	9.559675	
		PSO-ELM	100%	72.59%	5.1767	16.451451	
		GA-ELM	100%	70.37%	3.2609	6.280307	
		ELM	100%	68.15%		0.110669	
1:1	Hardlim()	CBQGA-ELM	100%	72%	0.086047	22.116859	20
		BQGA-ELM	93%	77.50%	0.0298	17.504254	
		QGA-ELM	25%	26%	1.1353	9.904138	
		PSO-ELM	97%	72.50%	0.0721	19.293127	
		GA-ELM	97.50%	88.50%	0.006	5.835251	
		ELM	81%	69.50%		0.117577	
	Sigmoid()	CBQGA-ELM	100%	66.50%	1.7298	22.322511	
		BQGA-ELM	100%	42%	0.5609	23.613902	
		QGA-ELM	100%	57%	0.4434	9.598749	
		PSO-ELM	100%	50%	2.2456	19.995686	
		GA-ELM	100%	47%	1.9359	6.398628	
		ELM	100%	49%		0.138599	

From Table 10: the hidden neurons' number is 10. When the activation function is Hardlim(), CBQGA-ELM obtained the best testing accuracy and validation RMSE than other diagnosis methods in different sample proportion, and the QGA-ELM diagnostic result is bad. When the activation function is Sigmoid (), BQGA-ELM obtained the better testing accuracy and validation RMSE than other diagnosis methods in different sample proportion.

As observed from Table 11, When activation function is Hardlim (), sample proportion is 3:1 and 2:1, and the hidden neurons' number is 20, CBQGA-ELM is best aiming at the validation RMSE, training and testing accuracy, and the QGA-ELM diagnostic result is less accuracy.

CBQGA-ELM is better aiming at the validation RMSE, training and testing accuracy than other diagnosis methods in Table 12, but diagnose result is worse compare with Table 10 and 11.

Overall, the optimized ELM variants deliver superior diagnostic accuracy relative to the standard ELM. However, they incur greater computational cost because they repeatedly execute parameter-tuning procedures. In addition, performance depends on a nontrivial interplay among the training sample ratio, the number of hidden neurons, and the chosen activation function—a relationship also illustrated in Figure 10.

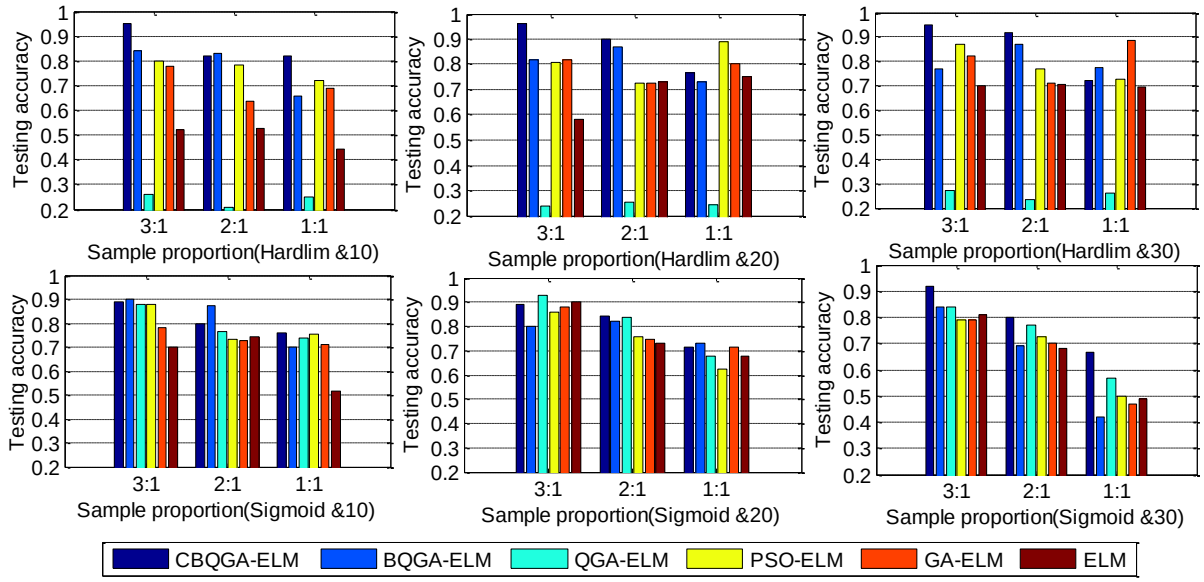

Figure 10: Bar of testing data diagnostic accuracy.

From Figure 10, when CBQGA-ELM has activation function, Hardlim(), the hidden neurons' number is 10 to 30, the sample proportion is 3:1, CBQGA-ELM is the best, and accuracy rate is greater than 90%. QGA does not adapt to optimize the ELM network whose activation function is Hardlim(); when the activation function is Hardlim(), the diagnosis effectiveness of several diagnosis methods does not show a significant decrease or rise follow the decrease of the sample proportion. But, when the activation function is Sigmoid(), the change trend of diagnosis effectiveness of the diagnosis methods is particularly obvious that diagnosis accuracy rate is decrease with decreasing the sample proportion as show in Figure 10 with the bottom three figures; under the Sigmoid() function and the sample proportion is 3:1, the number of BQGA-ELM neurons is10, the number of QGA-ELM neurons is 20 and the number of CBQGA-ELM neurons is 30, three methods' diagnosis accuracy can all reach 90% and above. Through the above analysis, CBQGA-ELM is more suitable for diagnosing rolling bearing fault samples than ELM optimized by other algorithms.

7.2 Error convergence

When the proportion sample is 3:1 and the activation function are Hardlim() and Sigmoid() respectively, Figures 11 and 12 present the validation error of some of diagnosis methods including CBQGA-ELM, BQGA-ELM, QGA-ELM, PSO-ELM and GA-ELM.

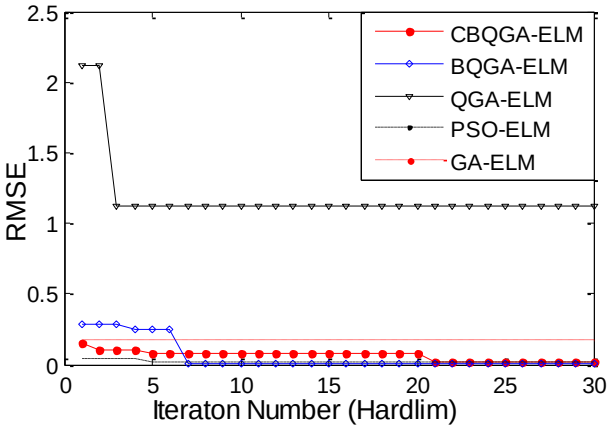

Figure 11: Average testing RMSE of different algorithms.

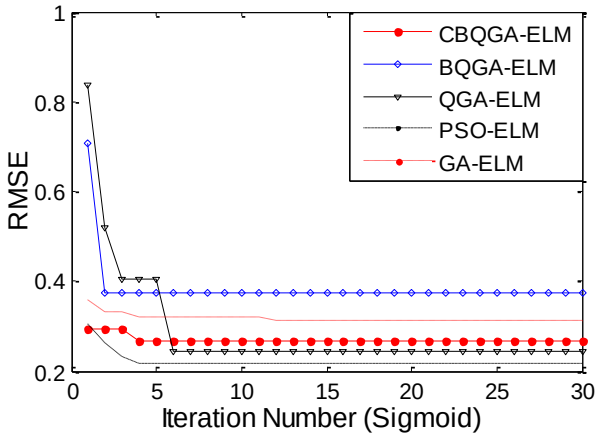

Figure 12: Average testing RMSE of different algorithms.

In general, when the activation function is Hardlim(), the overall effect of validation error convergence is poor than ELM's activation function is Sigmoid(). In the Figure 11, the top curves correspond to validation error (averaged over 30 trials) for QGA-ELM (Hardlim()), and it convergence effectiveness is poor due to QGA-ELM does not adapt to optimize the ELM network whose activation function is Hardlim(), so it diagnosis accuracy rate is also poor as shown in section 7.1. For other diagnosis methods, overall effect of validation error convergence is better. As observed from Figure 11 and 12, when activation function is Hardlim() and Sigmoid(), CBQGA-ELM can all obtain better results than other diagnosis methods. Through the analysis in section 7.1 and 7.2, CBQGA-ELM is more suitable for the diagnosis of rolling bearing fault samples than ELM optimized by other algorithms.

7.3 Computational time

In real aero-engine fault diagnosis, time is a critical constraint. A fast and accurate approach leaves greater margin for maintenance decisions and scheduling. Accordingly, this section trains all diagnostic methods across varying hidden-layer sizes and records their training times; the results are reported in Figure 13. In the figure, the bottom one curves are ELM network, due to lacks sophisticated computation for optimization, so it computational time is the least. The top two curves are CBQGA-ELM and BQGA-ELM, it computational time gradually increase with the increase of the number of hidden neurons. So selecting suitable neurons number is important, it can decrease computational time and ensure the diagnosis accuracy rate. Compared

with QGA-ELM, CBQGA-ELM and BQGA-ELM perform three-dimensional search, therefore computation time increase accordingly. (Note: the computational time is related to the computer configure, but it changes trend generally is the same.)

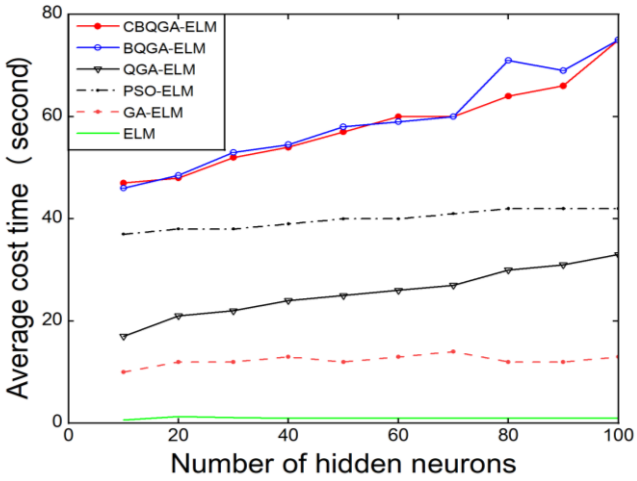

Figure 13: Average computational time (in second).

8 Conclusions

A new diagnostic method, CBQGA-ELM, is put forward. In this framework, the input weights (connections between input variables and hidden-layer neurons) and the hidden-layer biases are optimized by CBQGA, while the output weights are computed using the Moore–Penrose generalized inverse. The resulting optimized ELM is then applied to rolling bearing fault diagnosis.

To assess performance, CBQGA-ELM is evaluated on several real-world classification tasks and on rolling bearing fault diagnosis, and compared with alternative methods. These outcomes indicate that sample proportion, activation function, and the number of hidden neurons are critical design factors when using ELMs for classification. In particular, with Hardlim() or Sigmoid() activations, 10–30 hidden neurons, and a 3:1 training-to-testing sample proportion, CBQGA-ELM achieves the highest diagnostic accuracy for rolling bearing faults, outperforming the competing approaches. Detailed analyses in Section 7 further demonstrate that CBQGA-ELM is a more effective tool for rolling bearing fault diagnosis.

9 Conflicts of interest

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

Acknowledgements

The authors sincerely acknowledge their never-ending remembrance to Prof. PI Jun at Civil Aviation Administration of China. 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.

Author Contributions

Conceptualization, Yiqi Zhou., Hong Zhang.; methodology, Yiqi Zhou.; software, Sheng Ma.; validation, Yiqi Zhou., and Sheng Ma.; formal analysis, Yiqi Zhou and Hong Zhang.; investigation, Yiqi Zhou.; resources, Sheng Ma.; data curation, Zhengqing Zhu.; writing—original draft preparation, Yiqi Zhou.; writing—review and editing, Yiqi Zhou and Hong Zhang.; visualization, Yiqi Zhou. and Zhengqing Zhu; super-vision, Hong Zhang.; project administration, Yiqi Zhou and Hong Zhang.; funding acquisition, Hong Zhang. All authors have read and agreed to the published version of the manuscript.

Funding

The present work was supported by Fundamental Research Funds for Central Universities (3122019180).

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Data is contained within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

About The Author

Hong Zhang: (male, born 1978, Shanxi, China) is an Associate Professor in the Department of Engineering, Sino-European Institute of Aviation Engineering, Civil Aviation University of China (CAUC), Tianjin, China. He received the B.S. (2001) in Aircraft Power Engineering and the M.S. (2004) and Ph.D. (2010) in Aeronautics and Astronautics (Propulsion Theory and Engineering) from Nanjing University of Aeronautics and Astronautics. He has served at CAUC since 2013 (Associate Professor since 2019) and is Director of the China–Italy Joint Laboratory of Aviation (since 2023). His research interests include aero-engine structural vibration, fatigue and reliability; advanced structural design and failure analysis; and integrated aircraft–engine energy management. Email: h_zhang@cauc.edu.cn

Yiqi Zhou (male, born 2000, Hunan, China) is a master's candidate in Aeronautical Engineering at the Sino-European Institute of Aviation Engineering, Civil Aviation University of China (Sep. 2022–Jun. 2025). He received the B.Eng. in Aircraft Power Engineering from

the same institute in 2022. His research focuses on aero-engine blade vibration analysis and suppression. Email: 2022122053@cauc.edu.cn

Sheng Ma (born April 1992, Sichuan, China) is an Engineer at COMAC Software Co., Ltd. He holds a master's-level postgraduate education from the Civil Aviation University of China. His research interests include fault diagnosis and operational data analysis. Email: shifengzhijia@126.com

Zhengqing Zhu (male, born 2001, Shandong, China) is a Master's candidate in Aeronautical Engineering at the Sino-European Institute of Aviation Engineering, Civil Aviation University of China (Sep. 2023–Jun. 2026). He received the B.Eng. in Aircraft Power Engineering from the same institute in 2023. His research focuses on aircraft systems and parameter analysis and optimization for turbofan engine performance. Email: 2023122030@cauc.edu.cn

References

[1] C. Lv, J. Chang,, W. Bao, and D. Yu, "Recent research progress on airbreathing aero-engine control algorithm," Propulsion and Power Research, vol.11, no.1, pp. 1–57, 2022.

[2] P. Ma, H. Wang, D. Li, S. Yuan, and X. Fan., "Passive acoustic liners for aero-engine noise control: A review," Mechanics of Advanced Materials and Structures, vol. 31, no. 30, pp. 12834-12849, 2024.

[3] H. Zhao, L. Wang, Z. Zhao, and W. Deng, "A new fault diagnosis approach using parameterized time-reassigned multisynchrosqueezing transform for rolling bearings," IEEE transactions on reliability, vol. 74, no.1, 1, pp. 2363-2372, 2024.

[4] Y. Li., X. J. Xu, and Z. H. Huang, "Failure prediction for machine tools from the perspective of inverse problem," Engineering Computations, vol. 42, no. 5, pp. 1862-1879, 2025 .

[5] J. Li, Y. Li, J. Song, J. Zhang, and S Zhang, "Quantum support vector machine for classifying noisy data," IEEE Transactions on Computers, vol. 73, no. 9, pp. 2233-2247, 2024.

[6] S. Xia, X. Lian, G. Wang, et al., "Gbsvm: an efficient and robust support vector machine framework via granular-ball computing," IEEE Transactions on Neural Networks and Learning Systems, vol. 36, no. 5, pp. 9253-9267, 2024.

[7] J. Dong, Z. Wang, J. Wu, et al., "A novel runoff prediction model based on support vector machine and gate recurrent unit with secondary mode decomposition." Water Resources Management, vol. 38, no. 5, pp. 1655-1674, 2024.

[8] T. Ahmed Khan, et al., "Sentiment analysis using support vector machine and random forest," Journal of Informatics and Web Engineering, vol. 3, no. 1, pp. 67-75, 2024.

[9] J. Wang, Z. Li, et al., "Multiple kernel clustering with adaptive multi-scale partition selection," IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 11, pp. 6641-6652, 2024.

[10] G. L. Suryawanshi, S. K. Patil and R. G. Desavale, "Empirical-based DA and ANN to diagnose misalignment in rotor-bearing system, " Nondestructive Testing and Evaluation,

vol. 39, no. 4, pp. 776-801, 2024.

[11] S. Jeyakrishnan, et al., “An integration of RSM and ANN modelling approach for prediction of FSW joint properties in AA7178/AA5456 alloys, “ Canadian Metallurgical Quarterly, vol. 64, no. 1, pp. 43-60, 2025.

[12] J. Pi, J. Huang, and L. Ma, “Aeroengine fault diagnosis using optimized Elman neural network,” Mathematical Problems in Engineering, vol. 2017, no. 9, pp. 1–8, 2017.

[13] C.Satyanarayana, M. K. Yadav, and, M. Nath, “Multiquadric based RBF-HFD approximation formulas and convergence properties,” Engineering Analysis with Boundary Elements, vol. 160, no. 1, pp. 234-257, 2024.

[14] A. Wang, X. I. Yang, and M. Ovchinnikov, “An investigation of LES wall modeling for Rayleigh–Bénard convection via interpretable and physics-aware feedforward neural networks with DNS,” Journal of the Atmospheric Sciences, vol. 81, no. 2, pp. 435-458, 2024.

[15] Q.-Y. Zhu, A.-K. Qin, P. N. Suganthan, et al., “Evolutionary extreme learning machine,” Pattern Recognition, vol. 38, no. 10, pp. 1759–1763, 2005.

[16] M. A. A. Albadr, F. T. AL-Dhief, et al., “Online sequential extreme learning machine approach for breast cancer diagnosis,”Neural Computing and Applications, vol. 36, no. 18, pp. 10413-10429, 2024.

[17] X. Yang, S. Pang, W. Shen, et al., “Aero engine fault diagnosis using an optimized extreme learning machine,” International Journal of Aerospace Engineering, 2016, pp. 1–10, 2016.

[18] J. Lu, J. Huang, and F. Lu, “Sensor fault diagnosis for aero engine based on online sequential extreme learning machine with memory principle,” Energies, vol. 10, no. 1, pp. 39, 2017.

[19] Y. Zhang, J. Zhao, Y. Jia, and X. Shen, “An improved adaptive quantum genetic algorithm as classical optimizer for the quantum approximate optimization algorithm on MaxCut problem,” Quantum Information Processing, vol. 24, no. 7, pp. 1-30, 2025.

[20] R. S. Amal, and J. S Ivan, “A quantum genetic algorithm for optimization problems on the Bloch spher,” Quantum Information Processing, vol. 21, no. 2, pp. 43, 2022.

[21] S. S. Rosales, O. Montiel, et al., “Comparison of Performance of Amazon Braket Using a Quantum Genetic Algorithm,” Computación y Sistemas, vol. 28, no. 3, pp. 1429-1448, 2024.

[22] J. Hao, J. Mou, et al., “A novel color image encryption algorithm based on the fractional order laser chaotic system and the DNA mutation principle,” Multimedia Tools and Applications, vol. 81, no. 1, pp. 559-587, 2022.

[23] H. T. Choi, and J. H. Kim, “The $\mathscr{L}1$ controller synthesis for piecewise continuous nonlinear systems via set invariance principles,” International Journal of Robust and Nonlinear Control, vol. 33, no. 14, pp. 8670-8692, 2023.

[24] J. S. Manoharan, “Study of variants of extreme learning machine (ELM) brands and its performance measure on classification algorithm,” Journal of Soft Computing Paradigm (JSCP), vol. 3, no. 2, pp. 83-95, 2021.

[25] B. Alhijawi, and A. Awajan, “Genetic algorithms: Theory, genetic operators, solutions, and applications,” Evolutionary Intelligence, vol. 17, no. 3, pp. 1245-1256, 2024.

[26] X. Luo, J. Chen, Y. Yuan, and Z. Wang, “Pseudo gradient-adjusted particle swarm optimization for accurate adaptive latent factor analysis,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 54, no. 4, pp.2213-2226, 2024.

[27] M. Wang, Z. Cao, and Y. Li, “Quantum-Behaved Particle Swarm Optimization Based on Concentration Selection Probability Assignment Weights for Power System Economic Dispatch,” IEEJ Transactions on Electrical and Electronic Engineering, vol. 19, no. 5, pp. 640-652, 2024.