

Research Article

Fault Diagnosis for Engine Based on Single-Stage Extreme Learning Machine

Fei Gao and Jiangang Lv

Mechanical Engineering College, 97 West Heping Road, Shijiazhuang, Hebei Province 050003, China

Correspondence should be addressed to Fei Gao; flyinghigh6@126.com

Received 15 March 2016; Revised 6 August 2016; Accepted 29 August 2016

Academic Editor: Yan-Jun Liu

Copyright © 2016 F. Gao and J. Lv. 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.

Single-Stage Extreme Learning Machine (SS-ELM) is presented to dispose of the mechanical fault diagnosis in this paper. Based on it, the traditional mapping type of extreme learning machine (ELM) has been changed and the eigenvectors extracted from signal processing methods are directly regarded as outputs of the network's hidden layer. Then the uncertainty that training data transformed from the input space to the ELM feature space with the ELM mapping and problem of the selection of the hidden nodes are avoided effectively. The experiment results of diesel engine fault diagnosis show good performance of the SS-ELM algorithm.

1. Introduction

As representative equipment, the engine is general power source, and its safety and reliability is very important. In equipment fault diagnosis, reciprocating machinery faults are the most difficult cases. In order to solve the problem, piezoelectric pressure sensors, accelerometers, and sound sensors are widely used to measure signals from the engine. Faults are defined as the deviations from normal behaviors in the plant. Because the engine's working condition is so bad and its structure is so complex, signals of engine fault are nonstationary and nonlinear. It is difficult to extract a threshold which can clearly reflect the fault characteristics. So the methods which combine signal processing and intelligent pattern recognition are used to actualize fault diagnosis. Firstly, signal processing methods, such as frequency spectrum analysis, wavelet transform, Hilbert-Huang transform, and mathematical morphology, are used to denoise the measured signals and extract the feature vector which could broadly reflect the fault characteristics and types. Secondly, intelligent pattern recognition methods, such as artificial neural network and support vector machine (SVM), are used to map the feature vector into a higher dimensional feature space and classify the fault mode based on iterative optimization or statistical learning.

The combination of signal processing and intelligent pattern recognition solves the diagnosis puzzler of mechanical fault in a certain extent. But so many parameters need to be tuned to achieve a preferable fault classification rate when the recognition algorithm is applied. The computational complexity and computing time cost of the recognition also limit its effective realization in embedded systems. As a novel learning algorithm, ELM has been proposed recently by Huang et al. [1] for single hidden layer feedforward neural networks (SLFNs). Different from gradient-descent-based methods, ELM randomly chooses input weights (linking the input layer to the hidden layer) and hidden bias, and the output weights (linking the hidden layer to the output layer) are analytically determined using the Moore-Penrose generalized pseudoinverse instead of tuning them. Experimental results show that the learning speed of ELM can be thousands of times faster than gradient-descent learning algorithms. So it receives wide applications such as fault prognosis of mechanical components [2], fault classification in series compensated transmission line [3], fault diagnosis on hydraulic tube tester [4], and computer aided diagnosis system [5].

However, because of random mapping from the input space to some feature space, numerical stability of the output

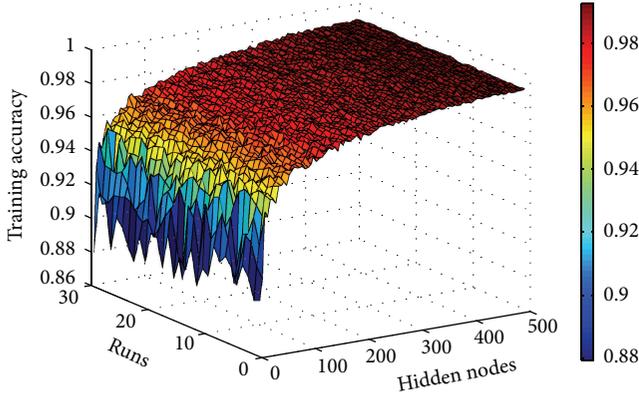


FIGURE 1: Training accuracy of ELM variation with respect to hidden nodes and initial parameters (dataset: Page Blocks).

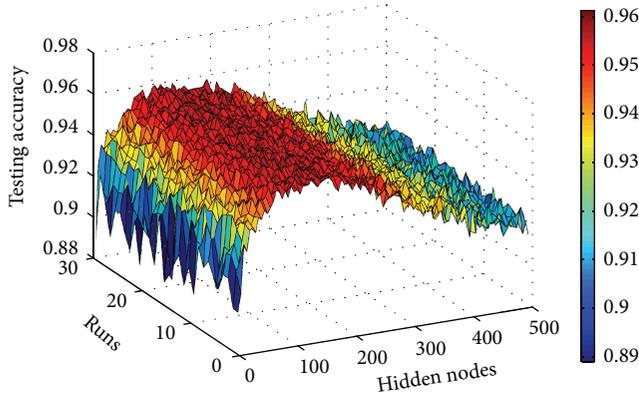


FIGURE 2: Testing accuracy of ELM variation with respect to hidden nodes and initial parameters (dataset: Page Blocks).

has been generally ignored. On the other hand, random selection of the input weights and biases results in a large number of hidden units which consumes so much computing time. To solve these shortages, many improved ELM algorithms were investigated recently [6–9]. Zhao et al. [10] proposed an input weight selection algorithm for an ELM with linear hidden nodes to improve the ill-conditioned problem. Huynh and Won proposed Least Square Extreme Learning Machine (LS-ELM) [11], Regularized Least Square Extreme Learning Machine (RLS-ELM) [12], and SVD-Neural classifier [13]. Projection Vector Machine was proposed by Deng et al. [14] for high-dimension small-sample data. Although these improved algorithms have enhanced numerical stability of the ELM output in a certain extent, some output fluctuations still exist. For many applications with high security requirements, unstable judgment may cause some fatal safety accidents, such as fault diagnosis for engine, industrial process control, control of the chaotic system, and operation condition monitoring of hydroelectric generating sets [15–18]. So, in order to introduce ELM to fault diagnosis of engine effectively, we proposed Single-Stage Extreme Learning Machine (SS-ELM) in this paper. Firstly, eigenvectors extracted from signal processing methods are directly regarded as the SS-ELM network's hidden layer output matrix. Secondly, the

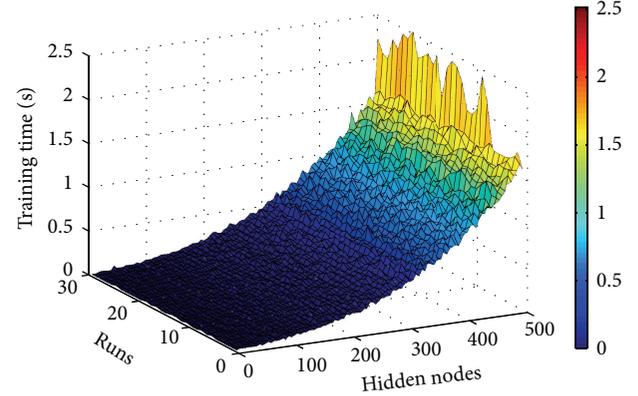


FIGURE 3: Training time of ELM variation with respect to hidden nodes and initial parameters (dataset: Page Blocks).

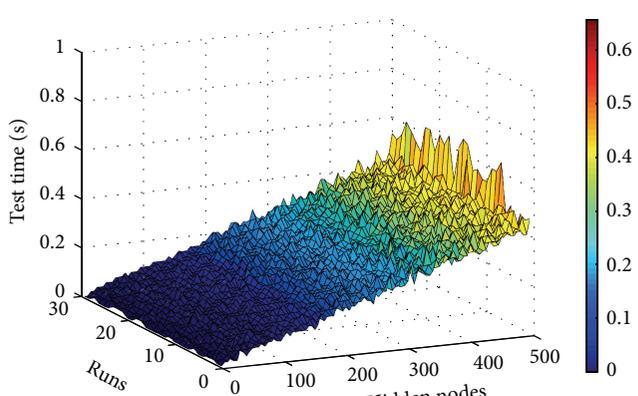


FIGURE 4: Testing time of ELM variation with respect to hidden nodes and initial parameters (dataset: Page Blocks).

Moore-Penrose generalized inverse is used to calculate the output weight. Then this method has lower computational complexity and could have been replanted to the embedded systems. Experimental results show that this approach is feasible in identifying engine faults.

The rest of this paper is organized as follows. Section 2 describes extreme learning machine algorithm and its shortage on classification. Single-Stage Extreme Learning Machine is presented in Section 3. Experimental results and analysis on engine fault diagnosis are shown in Section 4. Finally, conclusion is made in Section 5.

2. Extreme Learning Machine and Its Shortage on Classification

2.1. Single Hidden Layer Feedforward Networks. The standard architecture of single hidden layer feedforward networks consists of an input layer with d neurons, a hidden layer with N neurons, and an output layer with C neurons. Consider n arbitrary training samples $(\mathbf{x}_j, \mathbf{t}_j)_{j=1}^n$, where $\mathbf{x}_j = [x_{j1} \ x_{j2} \ \cdots \ x_{jd}]^T \in \mathbf{R}^d$ and $\mathbf{t}_j = [t_{j1} \ t_{j2} \ \cdots \ t_{jC}]^T \in \mathbf{R}^C$ are the j th input pattern and the corresponding desired

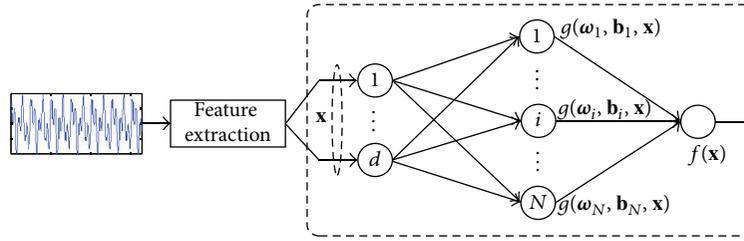


FIGURE 5: Methodology for extreme learning machine.

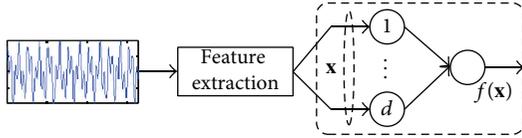


FIGURE 6: Methodology for Single-Stage Extreme Learning Machine.

output. Then SLFNs with activation function $g(\mathbf{x})$ can be mathematically modeled as

$$\mathbf{o}_j = \sum_{i=1}^N \beta_i g(\boldsymbol{\omega}_i \cdot \mathbf{x}_j + \mathbf{b}_i), \quad j = 1, \dots, n, \quad (1)$$

where $\boldsymbol{\omega}_i = [\omega_{i1}, \omega_{i2}, \dots, \omega_{id}]^T$ is the input weight vector connecting the input neurons and the i th hidden node, \mathbf{b}_i is the threshold of the i th hidden node, $\boldsymbol{\beta}_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{iC}]^T$ is the weight vector connecting the i th hidden node with the output neurons, \mathbf{o}_j is the real output vector of SLFNs, and $\boldsymbol{\omega}_i \cdot \mathbf{x}_j$ is the scalar product of $\boldsymbol{\omega}_i$ and \mathbf{x}_j .

So the main aim of training process is to minimize the following error function by adjusting the network parameters $\boldsymbol{\omega}_i$, \mathbf{b}_i , and $\boldsymbol{\beta}_i$, and the error function can be defined by

$$\xi = \sum_{j=1}^n (\mathbf{o}_j - \mathbf{t}_j)^2. \quad (2)$$

2.2. Extreme Learning Machine. Traditionally, gradient-descent algorithm is used to train the SLFNs, in which the set of W is iteratively tuned by

$$W_k = W_{k-1} - \eta \frac{\partial \xi}{\partial W}, \quad (3)$$

where W consists of parameters $\boldsymbol{\omega}_i$, \mathbf{b}_i , and $\boldsymbol{\beta}_i$ and η denotes the learning rate. As a popular training algorithm for feedforward neural networks based on gradient-descent, back-propagation (BP) learning algorithm has been used in various fields, in which parameters are adjusted with error propagation from the output layer to the input layer. However, it is clear that these algorithms have a slow learning rate, easily get overlearning, and stop at the local minimum.

Recently, an effective training algorithm for SLFNs was proposed by Huang et al. [1] and called ELM. According to Huang and Babri [19], SLFNs with at most N hidden nodes

and almost any nonlinear activation function can exactly learn n distinct observations. So if the standard SLFNs with N hidden nodes can approximate these n distinct observations with zero error, it implies that there exist $\boldsymbol{\omega}_i$, \mathbf{b}_i , and $\boldsymbol{\beta}_i$ such that

$$\sum_{i=1}^N \beta_i g(\boldsymbol{\omega}_i \cdot \mathbf{x}_j + \mathbf{b}_i) = \mathbf{t}_j. \quad (4)$$

Equation (4) can be written compactly as

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

where

$$\begin{aligned} \mathbf{H}(\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_N; \mathbf{b}_1, \dots, \mathbf{b}_N; \mathbf{x}_1, \dots, \mathbf{x}_n) \\ = \begin{bmatrix} g(\boldsymbol{\omega}_1, \mathbf{b}_1, \mathbf{x}_1) & \cdots & g(\boldsymbol{\omega}_N, \mathbf{b}_N, \mathbf{x}_1) \\ \vdots & \cdots & \vdots \\ g(\boldsymbol{\omega}_1, \mathbf{b}_1, \mathbf{x}_n) & \cdots & g(\boldsymbol{\omega}_N, \mathbf{b}_N, \mathbf{x}_n) \end{bmatrix}_{n \times N}, \\ \boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_N^T \end{bmatrix}_{N \times C}, \\ \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_n^T \end{bmatrix}_{n \times C}. \end{aligned} \quad (6)$$

As proposed by Huang et al. [1], \mathbf{H} is the hidden layer output matrix. The parameters $\boldsymbol{\omega}_i$ and \mathbf{b}_i (input weights and biases) may simply be assigned with random values and need not be adjusted during the training process. Then (5) becomes a linear system, due to the fact that the matrix \mathbf{H} may not always be square matrix, so the smallest norm least-squares of the network is estimated as

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^\dagger \mathbf{T}, \quad (7)$$

where \mathbf{H}^\dagger is the Moore-Penrose generalized inverse of matrix \mathbf{H} .

2.3. Shortage of Extreme Learning Machine on Classification. Although the ELM algorithm surmounts many puzzles, such

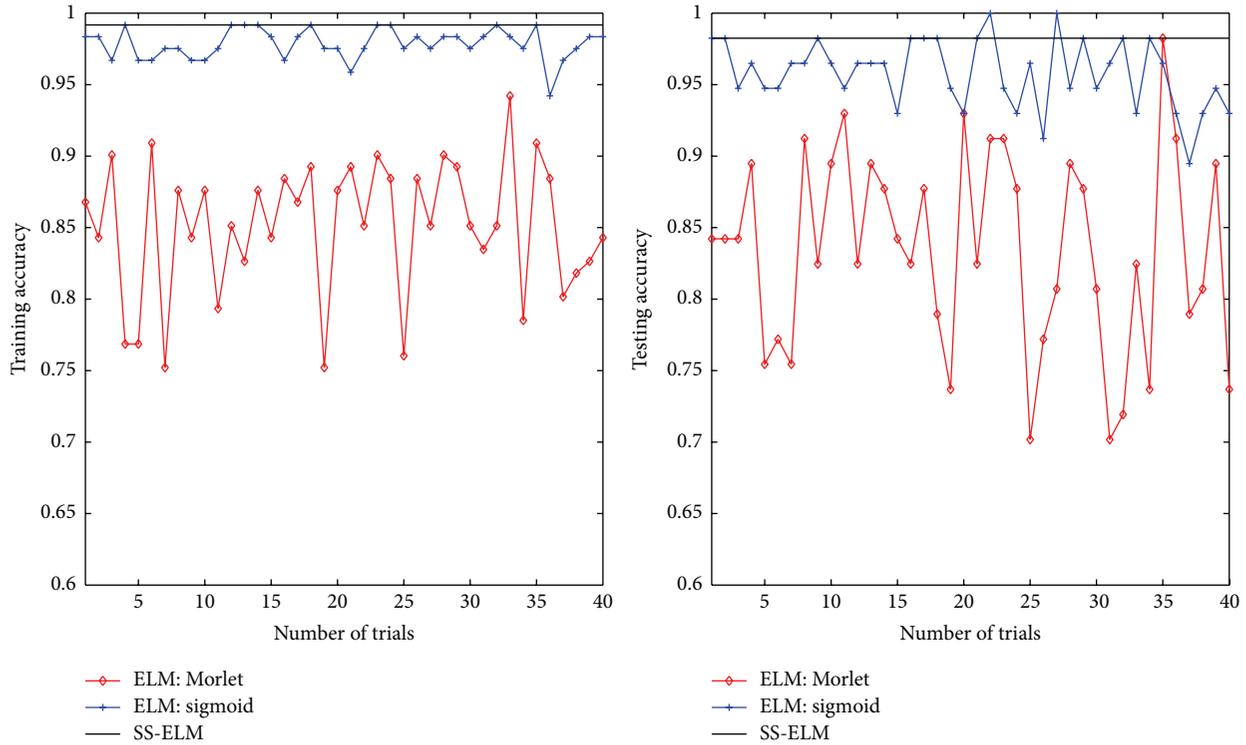


FIGURE 7: Training and testing accuracy of ELM and SS-ELM for Wine dataset.

as improper learning rate, overlearning, and local minima, which always lie in traditional gradient-descent approaches, random selection of input weights and biases may lead to an ill-conditioned problem so that the output of the network will be numerically unstable [10]. On the other hand, in order to get a preferable classification rate, the number of hidden nodes should be increased substantially which usually leads the complexity of network and training time to increase obviously.

In order to verify the problem of ELM with random mapping, we choose Page Blocks dataset from UCI Machine Learning Repository [20]. The initial number of hidden nodes is set as 50 and the incremental node for each simulation is set as 5. The sigmoidal function is chosen as the activation function; input weights and biases are determined randomly. Then the training and testing accuracy of ELM variations with respect to initial network parameters and hidden nodes on Satellite Image dataset are shown in Figures 1 and 2. Training and testing time of ELM with different hidden nodes are shown in Figures 3 and 4. The simulations are carried out in MATLAB 7.11.0 environment running in AMD, 2.2-GHZ CPU with 1 G RAM.

The simulation results show that different outputs of ELM were obtained with the same hidden nodes and activation function because of random selection of the input weights and biases. On the other hand, generalization performance of ELM depends on the proper selection of the input parameters, but it is difficult to search the best network parameters. It usually requires a large number of hidden nodes to achieve a preferable classification accuracy which results in slow response of the trained networks. According to Figure 3,

we know that the training time of ELM has an exponential manner approximately increasing along with the number of hidden nodes. In addition, the large number of hidden nodes also needs a large memory to store the parameters. These problems seriously restrict ELM's applications in embedded system.

For an ill-conditioned system, the change in the final solution may be large even if change in the output of ELM is small. So the numerical stability is an especially important aspect for fault diagnosis system. Equipments are maintained in a lagging manner and excess maintenance caused by underreporting and misjudgment of fault mode would result in serious accidents and larger financial burden. So ELM must be improved to be applied in the field of fault diagnosis because of the disturbance of the output.

3. Single-Stage Extreme Learning Machine

ELM algorithm can classify the multidimensional data which is extracted from the time domain measured signal. And the signal obtained from transducer is a kind of time series which cannot be imported to the ELM network directly. In other words, the ELM used three steps to accomplish classification of the time series. Firstly, some feature extraction methods are used to map the time series into feature space. The second step is to map the eigenvector from d -dimensional feature space into N -dimensional hidden layer space using the random input weights and hidden layer biases. The third step is to analytically determine the output weights by computing the pseudoinverse of the hidden layer output matrix.

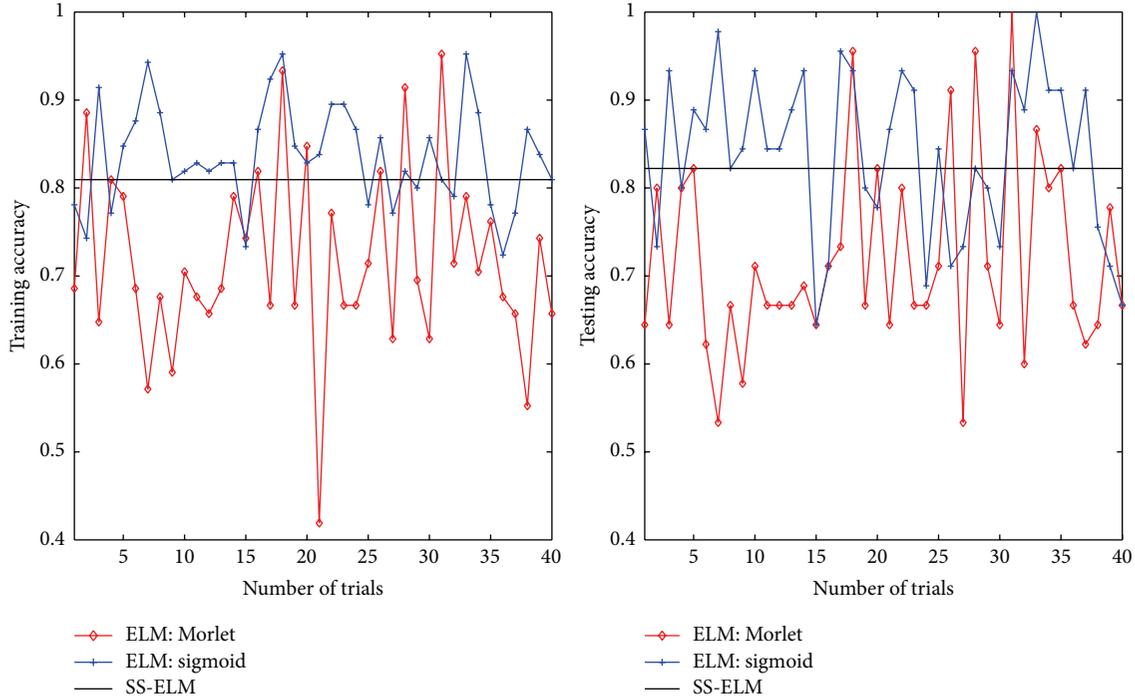


FIGURE 8: Training and testing accuracy of ELM and SS-ELM for Iris dataset.

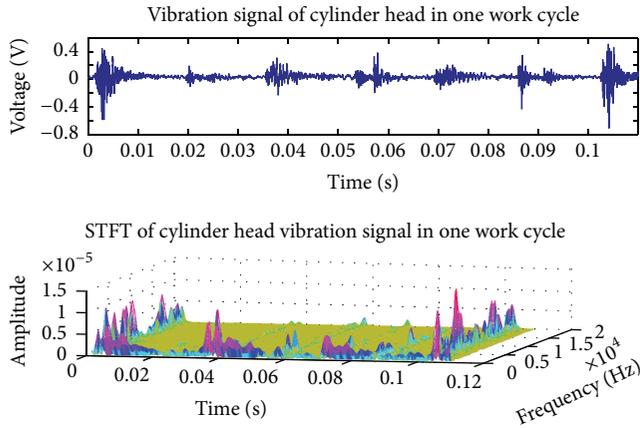


FIGURE 9: Vibration signal of cylinder head in one work cycle and its STFT.

According to the aforementioned content, we know that the random mapping results in the perturbation of output and a good classification performance requires much more hidden nodes. In other words, transformation of feature vectors increases the input dimension because the number of hidden nodes is always bigger than the number of input nodes. In order to solve the aforementioned problem, we predigest the network structure by regarding the feature vectors as the output of hidden layer; then the original network becomes a network only with two layers, the original random mapping is avoided, and original training process is simplified. So we call the improved structure Single-Stage Extreme Learning Machine (SS-ELM). In SS-ELM, the input matrix is equal to

the hidden layer output matrix in ELM and the output weight can be calculated by pseudoinverse operation of the input matrix. The sketch maps of ELM and SS-ELM are shown in Figures 5 and 6.

Given a set of training data $(\mathbf{x}_j, \mathbf{t}_j)_{j=1}^n$, where $\mathbf{x}_j = [x_{j1} \ x_{j2} \ \cdots \ x_{jd}]^T \in \mathbf{R}^d$ and $\mathbf{t}_j = [t_{j1} \ t_{j2} \ \cdots \ t_{jC}]^T \in \mathbf{R}^C$ are the j th input data and its desired output, then the input matrix \mathbf{X} can be defined by $\mathbf{X}_{n \times d} = [x_1, x_2, \dots, x_n]^T$ and the desired output matrix \mathbf{T} can be defined by $\mathbf{T}_{n \times C} = [t_1, t_2, \dots, t_n]^T$. The output weight of the network based on SS-ELM could be obtained by

$$\hat{\beta} = \mathbf{X}^\dagger \mathbf{T}, \quad (8)$$

where \mathbf{X}^\dagger is the Moore-Penrose generalized inverse of matrix \mathbf{X} .

The training times of SS-ELM and ELM are mainly costed by computing the Moore-Penrose generalized inverse of the hidden layer output matrix [21]. In most cases, singular value decomposition (SVD) is used to compute the Moore-Penrose generalized inverse; for the matrix $\mathbf{H} \in \mathbf{R}^{n \times N}$, the computational complexity of SVD is $O(4nN^2 + 8N^3)$ [22]. If the number of hidden layer nodes becomes large, computational time of SVD will remarkably rise. With a compact network structure, SS-ELM can achieve both better stability of output and lower computational complexity.

Consider a special structure of ELM in which the number of hidden nodes is equal to the dimension of input feature vector to analyze the random mapping from input space to hidden space. Then we investigate whether a difference exists between the special structure of ELM and SS-ELM or not.

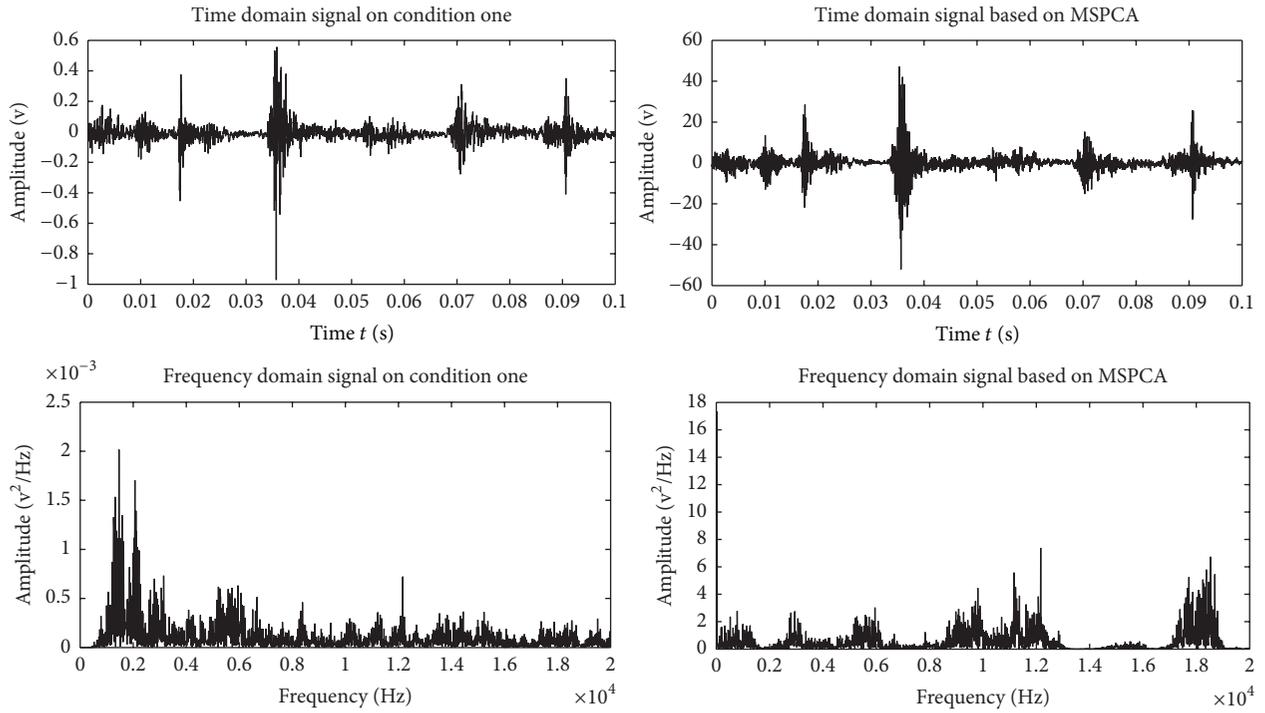


FIGURE 10: Signal in time domain and power spectrum on normal condition.

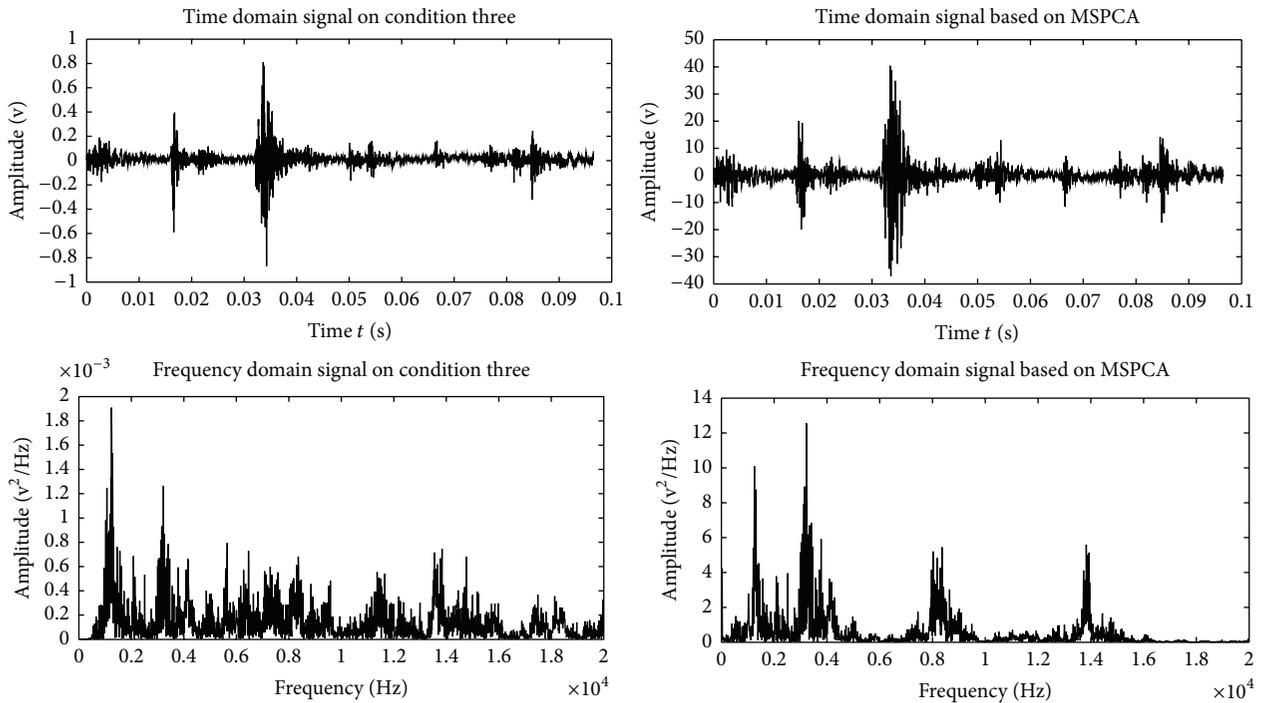


FIGURE 11: Signal in time domain and power spectrum on second cylinder misfire condition.

Performance comparison of SS-ELM and the ELM algorithm for Iris and Wine dataset is carried out. 70% and 30% of samples in the dataset are randomly chosen as train and test data at each trial. The details of datasets and the average learning time of forty trials for these two algorithms are

listed in Table 1. Training and testing accuracy of 40 trials of simulations for each method are shown in Figures 7 and 8.

The aforementioned simulation results show that the learning speed of SS-ELM is as fast as ELM. And the main difference between SS-ELM and ELM algorithms lies in the

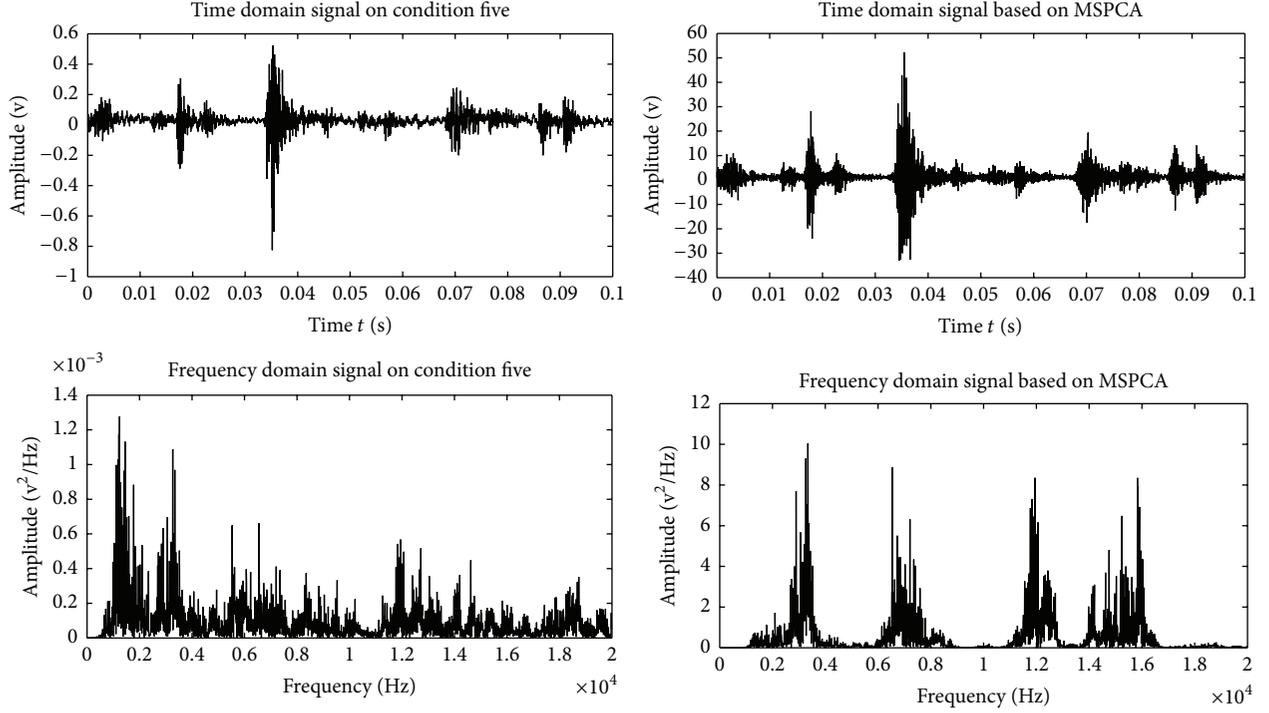


FIGURE 12: Signal in time domain and power spectrum on small exhaust valve clearance condition.

TABLE 1: Details of real-world applications and average learning time for each.

	Number of samples	Number of attributes	Number of classes	ELM		SS-ELM	
				Training time/s	Testing time/s	Training time/s	Testing time/s
Wine	178	13	3	0.0016	0.0008	0.0012	0.0004
Iris	150	4	3	0.0012	0.0008	0.0004	0.0004

stability of the output of the network. ELM has an unstable output due to the random mapping from original space to the hidden space, and SS-ELM has a robust output due to the simplified network structure.

In some real-world applications, the training data may be chunk-by-chunk or one-by-one. Under these circumstances, the incremental learning algorithms may outperform batch learning algorithms as incremental learning algorithms do not require retraining the old data whenever new data is received [23]. Therefore, the batch SS-ELM algorithm can also be extended to accommodate online incremental learning to the online problems.

Given a set of initial training data $S_0 = (\mathbf{x}_j, \mathbf{t}_j)_{j=1}^{n_0}$, the initial input matrix \mathbf{X}_0 and output matrix \mathbf{T}_0 could be got easily, so the initial output weight can be obtained by

$$\hat{\boldsymbol{\beta}}^{(0)} = \mathbf{X}_0^\dagger \mathbf{T}_0. \quad (9)$$

If $\mathbf{X}_0^T \mathbf{X}_0$ tends to become nonsingular, the initial output weight can be define by

$$\hat{\boldsymbol{\beta}}^{(0)} = (\mathbf{X}_0^T \mathbf{X}_0)^{-1} \mathbf{X}_0^T \mathbf{T}_0 = \mathbf{K}_0^{-1} \mathbf{X}_0^T \mathbf{T}_0, \quad (10)$$

where $\mathbf{K}_0 = \mathbf{X}_0^T \mathbf{X}_0$. If $\mathbf{X}_0^T \mathbf{X}_0$ tends to become singular, one can make it nonsingular by adding a constant diagonal matrix; \mathbf{K}_0 can be redefined as $\mathbf{K}_0 = \mathbf{X}_0^T \mathbf{X}_0 + \lambda \mathbf{I}$.

When other n_1 observations of the second training subset $S_1 = (\mathbf{x}_j, \mathbf{t}_j)_{j=n_0+1}^{n_0+n_1}$ are received, the training problem becomes minimizing

$$\left\| \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \end{bmatrix} \boldsymbol{\beta} - \begin{bmatrix} \mathbf{T}_0 \\ \mathbf{T}_1 \end{bmatrix} \right\|, \quad (11)$$

where $\mathbf{X}_1 = [\mathbf{x}_{n_0+1}, \mathbf{x}_{n_0+2}, \dots, \mathbf{x}_{n_0+n_1}]^T$, $\mathbf{T}_1 = [\mathbf{t}_{n_0+1}, \mathbf{t}_{n_0+2}, \dots, \mathbf{t}_{n_0+n_1}]^T$.

Considering both blocks of training sets S_0 and S_1 , the output weight $\boldsymbol{\beta}$ becomes

$$\begin{aligned} \hat{\boldsymbol{\beta}}^{(1)} &= \left[\begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \end{bmatrix}^T \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \end{bmatrix} \right]^{-1} \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \end{bmatrix}^T \begin{bmatrix} \mathbf{T}_0 \\ \mathbf{T}_1 \end{bmatrix} \\ &= (\mathbf{X}_0^T \mathbf{X}_0 + \mathbf{X}_1^T \mathbf{X}_1)^{-1} (\mathbf{X}_0^T \mathbf{T}_0 + \mathbf{X}_1^T \mathbf{T}_1) \\ &= \mathbf{K}_1^{-1} (\mathbf{K}_0 \hat{\boldsymbol{\beta}}^{(0)} + \mathbf{X}_1^T \mathbf{T}_1) \end{aligned}$$

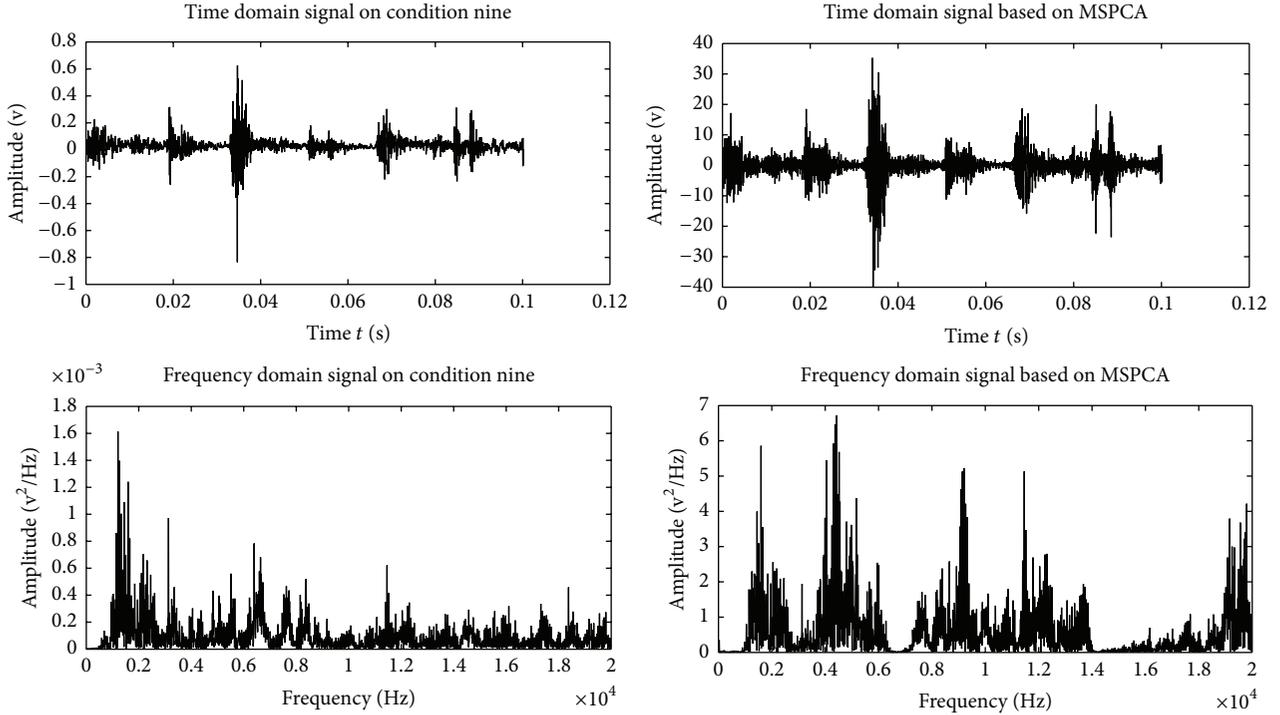


FIGURE 13: Signal in time domain and power spectrum on gas leak air supply valve condition.

$$\begin{aligned}
 &= \mathbf{K}_1^{-1} \left[(\mathbf{K}_1 - \mathbf{X}_1^T \mathbf{X}_1) \hat{\boldsymbol{\beta}}^{(0)} + \mathbf{X}_1^T \mathbf{T}_1 \right] \\
 &= \hat{\boldsymbol{\beta}}^{(0)} + \mathbf{K}_1^{-1} \mathbf{X}_1^T (\mathbf{T}_1 - \mathbf{X}_1 \hat{\boldsymbol{\beta}}^{(0)}),
 \end{aligned} \tag{12}$$

where $\mathbf{K}_1 = \mathbf{K}_0 + \mathbf{X}_1^T \mathbf{X}_1$.

4. Experimental Results

In this paper, we measured vibration signals on the F3L912-type diesel engine which consists of 3 cylinders and 4 strokes. The rotating speed is 1200 r/min and the engine is empty loaded when sampling acceleration signals on the first cylinder. The oil pressure signal can be used as indication to the vibration signal on cylinder head, so the oil pressure signal on the third cylinder is measured by the clip oil pressure sensor with vibration signal in the synchronization manner. The sample rate is 40 KHz in the experiment. We measured signals on eleven working conditions: (1) normal condition, (2) first cylinder misfire condition, (3) second cylinder misfire condition, (4) large exhaust valve clearance condition, (5) small exhaust valve condition, (6) large air supply valve clearance condition, (7) small air supply valve clearance condition, (8) gas leak exhaust valve condition, (9) gas leak air supply valve condition, (10) light oil leak condition, and (11) both first and second cylinder misfire condition. Vibration signal of cylinder head in one work cycle on normal condition and its Short-Time Fourier Transform (STFT) are shown in Figure 9.

As the aforementioned content, the vibration signal on cylinder head reflects the action time and intensity of five main excitation instances. From the time-frequency domain

distribution, we know that the vibration signal has a wide frequency band and it is nonstationary. Multiscale principal component analysis (MSPCA) [24] computes the PCA of wavelet coefficients at each scale and combines the results at relevant scales. Because of the multiscale nature, it has wide applications. So, in this paper, we used the MSPCA method to analyze the vibration signal on cylinder head. The original vibration signal on cylinder head and transformation based on MSPCA on four conditions are shown in Figures 10, 11, 12, and 13.

The wavelet packet can focus energy distribution of the signal on different decomposition coefficients, and it has advantages in feature extraction [25]. So we decompose the vibration signal to three layers using the wavelet function Daubechies 4 and reconstruct the conjugate filter coefficients of wavelet packet decomposition firstly. Then we calculate the energy of signal in each frequency band using Parseval theory and normalize the energy of each frequency band as the feature vector. 200 and 125 feature vectors in fault diagnosis problem are randomly chosen for training and testing. Forty trials for each algorithm are conducted, and then we take average of forty trials as final results. The training and testing accuracy are shown in Figure 14. The training time based on ELM with Morlet function and sigmoid function is 4.3 ms and 3.9 ms, in which the number of hidden layer nodes is equal to the dimension of feature vector. The training time of SS-ELM is 2.3 ms.

Experiment shows that the above mentioned diesel engine fault diagnosis method based on MSPCA and SS-ELM has a high accuracy rate for fault identification. SS-ELM overcomes the shortcoming that input vector must be

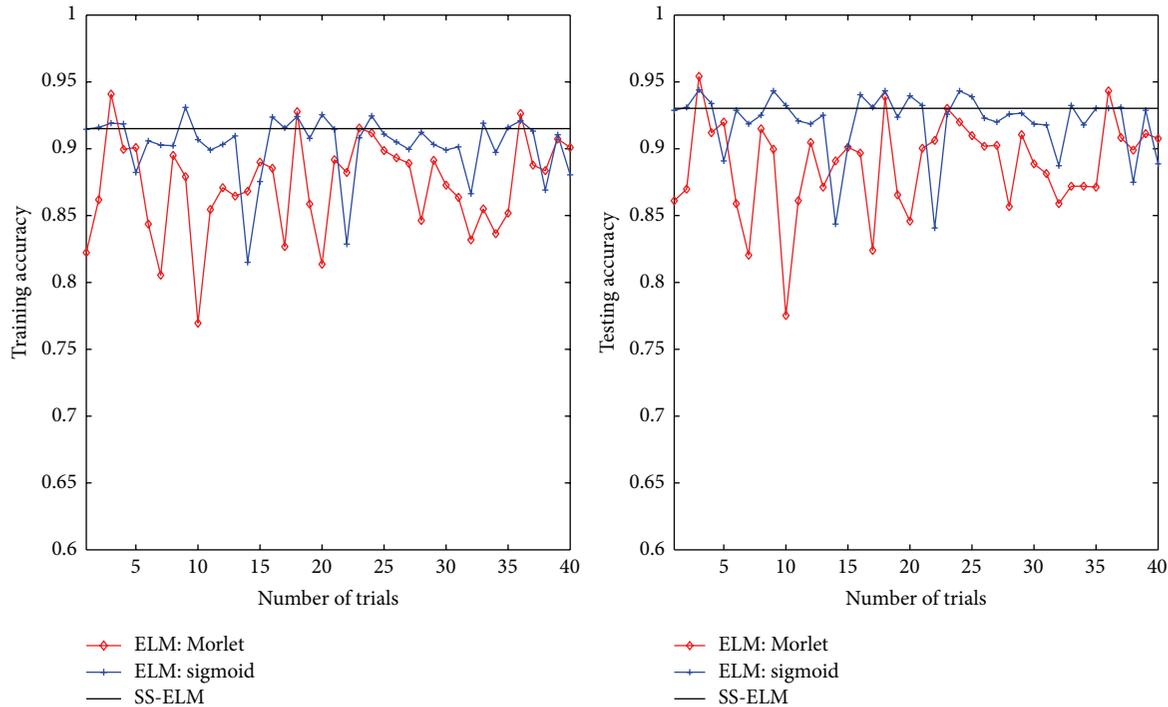


FIGURE 14: Fault identification results on engine based different algorithm.

randomly mapped to the kernel space of the hidden layer in ELM. Then SS-ELM avoids the problem that equipments are maintained in a lagging manner and excess maintenance caused by underreporting and misjudgment of fault mode. So we think that the SS-ELM can be used in the field of fault diagnosis.

5. Conclusion

In this paper, a fault diagnosis method for engine is proposed. Firstly, due to the random mapping from the input space to the hidden layer space of the traditional ELM, unstable output result which may cause some fatal safety accidents for fault diagnosis of engine exists. Then we simplify the original structure of network and propose the Single-Stage Extreme Learning Machine (SS-ELM). In SS-ELM, the original random mapping is avoided, original training process is simplified, the input matrix is equal to the hidden layer output matrix in ELM, and the output weight is calculated by pseudoinverse operation of the input matrix. Experimental results show that the output stability of the modified method outperforms the traditional ELM in which the input and hidden layers have the same dimension. The learning speed of SS-ELM is as fast as ELM, but the modified algorithm does not need to tune any parameter in the whole training process.

Competing Interests

The authors declare that they have no competing interests.

References

- [1] 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.
- [2] D. Martinez-Rego, O. Fontenla-Romero, B. Perez-Sanchez et al., "Fault prognosis of mechanical components using on-line learning neural networks," in *Artificial Neural Networks—ICANN 2010: 20th International Conference, Thessaloniki, Greece, September 15–18, 2010, Proceedings, Part I*, vol. 6352, no. 1 of *Lecture Notes in Computer Science*, pp. 60–66, Springer, Berlin, Germany, 2010.
- [3] V. Malathi, N. S. Marimuthu, and S. Baskar, "A comprehensive evaluation of multicategory classification methods for fault classification in series compensated transmission line," *Neural Computing & Applications*, vol. 19, no. 4, pp. 595–600, 2010.
- [4] X.-F. Hu, Z. Zhao, S. Wang, F.-L. Wang, D.-K. He, and S.-K. Wu, "Multi-stage extreme learning machine for fault diagnosis on hydraulic tube tester," *Neural Computing & Applications*, vol. 17, no. 4, pp. 399–403, 2008.
- [5] M. Gomathi and P. Thangaraj, "A computer aided diagnosis system for lung cancer detection using machine learning technique," *European Journal of Scientific Research*, vol. 51, no. 2, pp. 260–275, 2011.
- [6] N. Wang, J.-C. Sun, M. J. Er, and Y.-C. Liu, "Hybrid recursive least squares algorithm for online sequential identification using data chunks," *Neurocomputing*, vol. 174, pp. 651–660, 2016.
- [7] N. Wang, M. J. Er, and M. Han, "Generalized single-hidden layer feedforward networks for regression problems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 6, pp. 1161–1176, 2015.

- [8] N. Wang, M. J. Er, and M. Han, "Parsimonious extreme learning machine using recursive orthogonal least squares," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 10, pp. 1828–1841, 2014.
- [9] Z. Shao, M. J. Er, and N. Wang, "An efficient leave-one-out cross-validation-based extreme learning machine (ELOO-ELM) with minimal user intervention," *IEEE Transactions on Cybernetics*, vol. 46, no. 8, pp. 1939–1951, 2016.
- [10] G. Zhao, Z. Shen, and Z. Man, "Robust input weight selection for well-conditioned extreme learning machine," *Journal of Information Technology*, vol. 17, no. 1, pp. 1–18, 2011.
- [11] H. T. Huynh and Y. Won, "Small number of hidden units for ELM with two-stage linear model," *IEICE Transactions on Information and Systems*, vol. 91, no. 4, pp. 1042–1049, 2008.
- [12] H. T. Huynh, Y. Won, and J.-J. Kim, "An improvement of extreme learning machine for compact single-hidden-layer feedforward neural networks," *International Journal of Neural Systems*, vol. 18, no. 5, pp. 433–441, 2008.
- [13] H. T. Huynh and Y. Won, "Training single hidden layer feedforward neural networks by using singular value decomposition," in *Proceedings of the International Conference on Computer Sciences and Convergence Information Technology (ICCIT '09)*, pp. 1300–1304, Seoul, Republic of Korea, 2009.
- [14] W. Deng, Q. Zheng, S. Lian et al., "Projection vector machine: one-stage learning algorithm from high-dimension small-sample data," in *Proceedings of the IEEE International Joint Conference on Neural Network*, pp. 3375–3382, Barcelona, Spain, 2010.
- [15] X.-W. Chen, J.-G. Zhang, and Y.-J. Liu, "Research on the intelligent control and simulation of automobile cruise system based on fuzzy system," *Mathematical Problems in Engineering*, vol. 2016, Article ID 9760653, 12 pages, 2016.
- [16] G.-X. Wen, C. L. P. Chen, Y.-J. Liu, and Z. Liu, "Neural-network-based adaptive leader-following consensus control for second-order non-linear multi-agent systems," *IET Control Theory & Applications*, vol. 9, no. 13, pp. 1927–1934, 2015.
- [17] C. L. P. Chen, Y.-J. Liu, and G.-X. Wen, "Fuzzy neural network-based adaptive control for a class of uncertain nonlinear stochastic systems," *IEEE Transactions on Cybernetics*, vol. 44, no. 5, pp. 583–593, 2014.
- [18] Y. Gao and Y.-J. Liu, "Adaptive fuzzy optimal control using direct heuristic dynamic programming for chaotic discrete-time system," *Journal of Vibration and Control*, vol. 22, no. 2, pp. 595–603, 2016.
- [19] G.-B. Huang and H. A. Babri, "Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions," *IEEE Transactions on Neural Networks*, vol. 9, no. 1, pp. 224–229, 1998.
- [20] C. L. Blake and C. J. Merz, *UCI Repository of Machine Learning Databases*, Department of Information and Computer Sciences, University of California, Irvine, Calif, USA, <http://www.ics.uci.edu/~mllearn/MLRepository.html>.
- [21] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.
- [22] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Md, USA, 3rd edition, 1996.
- [23] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, 2006.
- [24] B. R. Bakshi, "Multiscale PCA with application to multivariate statistical process monitoring," *AIChE Journal*, vol. 44, no. 7, pp. 1596–1610, 1998.
- [25] C. S. Burrus, R. A. Goponath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms*, China Machine Press, 2008.



Hindawi

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

