

## Research Article

# A Novel Multimode Fault Classification Method Based on Deep Learning

Funa Zhou,<sup>1</sup> Yulin Gao,<sup>1</sup> and Chenglin Wen<sup>2</sup>

<sup>1</sup>*School of Computer and Information Engineering, Henan University, Kaifeng, China*

<sup>2</sup>*School of Automation, Hangzhou Dianzi University, Hangzhou, China*

Correspondence should be addressed to Yulin Gao; [gaoyulinhn@163.com](mailto:gaoyulinhn@163.com) and Chenglin Wen; [wencl@hdu.edu.cn](mailto:wencl@hdu.edu.cn)

Received 23 December 2016; Accepted 22 February 2017; Published 20 March 2017

Academic Editor: Youqing Wang

Copyright © 2017 Funa Zhou 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.

Due to the problem of load varying or environment changing, machinery equipment often operates in multimode. The data feature involved in the observation often varies with mode changing. Mode partition is a fundamental step before fault classification. This paper proposes a multimode classification method based on deep learning by constructing a hierarchical DNN model with the first hierarchy specially devised for the purpose of mode partition. In the second hierarchy, different DNN classification models are constructed for each mode to get more accurate fault classification result. For the purpose of providing helpful information for predictive maintenance, an additional DNN is constructed in the third hierarchy to further classify a certain fault in a given mode into several classes with different fault severity. The application to multimode fault classification of rolling bearing fault shows the effectiveness of the proposed method.

## 1. Introduction

Rolling bearing is a very pivotal component in rotating machines, which are widely used in large-scale automated industrial equipment. Mechanical failure caused by rolling bearings may cause abnormality of the rotating machinery system, resulting in huge economic losses, and even cause some unnecessary casualties [1–5]. Therefore, timely and precisely classification is critical for bearing monitoring.

The methods for mechanical equipment fault classification can be divided into qualitative model based method, quantitative model based method, and data-driven based method [6, 7]. Qualitative model and quantitative model based methods require precise mathematical model or a large amount of expert knowledge of the system, which will inevitably limit its application in fault classification field. In the recent two decades, data-driven method is widely used in fault detection of complex system. Instead of much more prior knowledge, data-driven approach can detect fault only through the measured data of the complex system [8–11]. The most common used data-driven fault classification methods are statistical feature extraction based methods and machine

learning based methods. However, the method based on statistical feature extraction can only realize fault detection and it is unable to realize fault classification. For fault classification, we had better use machine learning method such as Support Vector Machine (SVM) and artificial neural network (ANN).

In the field of mechanical system fault classification, because of the sensitivity of vibration spectrum to equipment failure, vibration signals are usually used as the data source for fault classification of mechanical equipment. Due to mechanical equipment's characteristics of being nonstable, nonlinear, large-scale, high-dimensional, and noise polluted, it is usually very difficult for precise fault feature extraction which is the most critical factor of the accuracy of mechanical equipment monitoring [12–14]. Some scholars have put forward some feature extraction methods that combine signal processing technology with machine learning method for fault classification of mechanical equipment. Widodo and Yang extract the frequency-domain feature as the data source of SVM to detect the machinery fault [13]. When the number of samples is small and the signals are nonstationary, Yu et al. proposed a bearing fault classification method by combining SVM

and Empirical Mode Decomposition (EMD) [10]. Hu et al. extracted the energy of each wavelet packet transform (WPT) node as the preextracted feature to develop a combined WPT-SVM based method for more accurate bearing fault classification [15]. Wang et al. also used WPT to extract nonstationary characteristics of the bearing's vibration signal as the preextracted feature of ANN [16]. The method uses the nonlinear learning classification ability and self-organizing ability of ANN to classify and diagnose bearing faults. Yang and Tang proposed a method combining expert system and back propagation neural network (BPNN) [17]. This method makes full use of the advantages of expert system and ANN to successfully detect the bearing failure. Since bearing vibration signals are susceptible to Gaussian noise, Jiang et al. used high level statistics as the feature vector of BPNN to improve the performance of BPNN in bearing fault classification [18]. However, SVM and BPNN share the shortcomings of shallow learning method: SVM is an algorithm of two classifiers, and it is inefficient in multiclassification especially in the case when the sample number of observation is very large. Selecting the appropriate kernel function and scale parameter usually needs a wealth of experience. ANN also suffers from many defects, such as the following: (1) ANN has a slow convergence rate and can easily converge to the local optimum and (2) ANN is ineffective in feature learning of complex nonlinear data and usually results in poor classification accuracy. In summary, SVM and BPNN as the shallow learning methods could not well extract the data feature involved in the high-dimensional unsteady data [19]. With the load varying, bearing can work in different steady state, which is called "multimode" phenomenon. Current research work on machine learning based classification did not take multimode problem into account.

For multimode process, the data feature of each mode is different [20], but current research on bearing fault classification usually regards it as a single mode for simplicity of data processing which will result in inaccurate classification result since feature extracted is inaccurate [21–23]. Therefore, mode partition should be implemented before fault feature extraction of a separate mode for accurate feature extraction. Zhang et al. proposed an improved  $k$ -means clustering algorithm based on existing modal partition method [20]. Song et al. studied the issue to distinguish stability mode from transition mode without the number of modes known in advance [24]. Zhao et al. separated multiple modalities according to the diversity analysis in operational phases and established online monitoring method along multiple batch directions [25]. Zhang et al. used modal subspace separation method to deal with multimode monitoring problems [26]. By using various characteristics of the subspace, different mode can well be separated, which can provide chance for more accurate multimode fault classification.

Unfortunately, mode partition and corresponding fault monitoring method for certain multimode processes are only specially developed for a specific industrial process [20, 24–27]. It is required to develop a more universal method. Deep learning is a promising ubiquitous feature extraction tool which has attracted wide attention by scholars from various fields [21, 28–30]. Comparing to shallow learning,

deep learning can well process the feature extraction and the issue of nonlinear big data by constructing a deep network [31, 32]. Through the unsupervised layer-by-layer greedy training algorithm and BP-based global parameter fine-tuning, deep neural network (DNN) can not only avoid the local optimization problem, but also solve the problem of limitation in number of labeled samples and the limitation in generalization ability. Deep learning method was firstly proposed by Hinton and Salakhutdinov in 2006 [22]. In view of its excellent feature extraction capabilities, it also attracts the attention of fault classification experts. Lu et al. successfully used the better feature extraction ability of deep neural network to diagnose the bearing fault [33]. The proposed method overcomes the shortcomings that the traditional feature extraction method could not discover the unknown type fault timely and effectively. Jia et al. used deep neural network to monitor the failure of bearings [34]. Gan et al. proposed a fault classification method based on hierarchical neural network [11]. By constructing a two-layer neural network, the method not only could locate the position of bearing fault but also effectively mines the fault size of the bearing in the same position. Deep learning, as one of the most popular machine learning methods, has brought a subversive revolution to the field of artificial intelligence. However, application about the deep learning is still in infancy, during the application process; there are also many issues demanding improvement. For example, the data in [11] are derived from a single mode, without considering the multimode observation caused by load varying problem. Therefore, it cannot fully extract the fault feature involved in the observation of different mode which is essential for the accuracy of multimode fault classification.

To solve the above-mentioned problems, this paper presents a multimode fault classification method based on deep learning. First, a DNN model is constructed, and the trained network is used to mode partition; then, a new set of DNNs are constructed for observation data of each mode, and the trained networks are used to determine which component fails to implement fault location recognition; finally, for a certain fault in a given mode, another DNN is constructed to classify those observation data with different fault size.

The remainder of this paper is as follows: Section 2 overviews the theory of deep learning. Section 3 develops a multimode fault classification method based on DNN by hierarchically constructing DNN models with different purpose. In Section 4, effectiveness of the proposed multimode fault classification method is demonstrated by experiments analysis. Section 5 concludes this paper.

## 2. Theory of Deep Learning

Deep learning is a method based on unsupervised feature learning. We use deep learning theory to construct DNN. DNN training process consists of two steps: (1) using the unsupervised learning algorithm to pretrain the network layer by layer, which is helpful for DNN to efficiently mine features from raw data; (2) using the back propagation algorithm to fine-tune the parameters of the whole network,

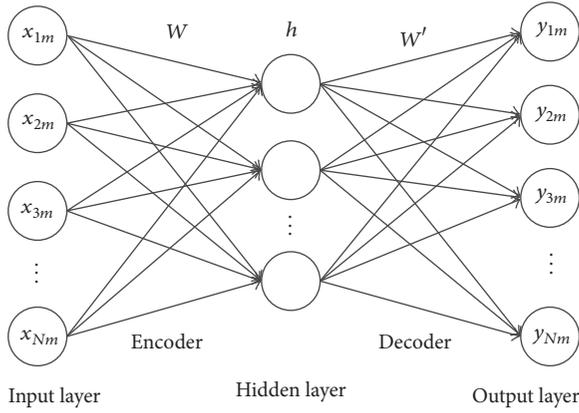


FIGURE 1: The model of AutoEncoder.

optimizing the performance of DNN to mine raw feature. In this paper, DNN is pretrained by multistacking AutoEncoder (AE).

**2.1. AutoEncoder.** AutoEncoder is an unsupervised machine learning structure, and it can be viewed as a three-layer forward artificial neural network, as shown in Figure 1. It consists of the input layer, the hidden layer, and the output layer. AutoEncoder is a very special neural network with single hidden layer, whose output is equal to the input. AutoEncoder network parameters can be adjusted by repeated training process, such that the reconstructed output is an approximation with high accuracy of the input. AutoEncoder is composed of two parts: encoder and decoder. The encoder network encodes the input data from the high-dimensional space into low-dimensional space; then the low-dimensional space data is mapped into high-dimensional space through decoder network which realized the reconstruction process from output to input. Therefore, the low-dimensional space data can be used as the characteristic representation of the input data.

Given an unlabeled dataset  $\{x_{nm}\}$ , ( $n = 1, 2, \dots, N$ ;  $m = 1, 2, \dots, M$ ) consisting of  $N$  observation features or variables, each observation variable has  $M$  samples. The encoder network encodes the sample  $x_m = [x_{1m}, x_{2m}, \dots, x_{Nm}]^T$  to the hidden activate value  $h$  with an activation function  $f_\theta$ . The encoder process is described as follows:

$$h = f_\theta(x_m) = \sigma(Wx_m + b), \quad (1)$$

where  $f_\theta$  is the encoder function, Sigmoid function  $\sigma$  is usually taken as the activation function in the encoder process,  $W$  is the weight matrix of the network between input layer and the hidden layer,  $b$  is the bias vector generated by the encoder network, and  $\theta = \{W, b\}$  is the connection parameter between the input layer and the hidden layer. The Sigmoid function can be depicted via

$$\sigma(x) = \frac{1}{[1 + \exp(-x)]}. \quad (2)$$

Similarly, for the decoder network, the feature matrix  $h$  obtained from encoder network is used to reconstruct  $y_m$

through the decoder network such that the reconstructed  $y_m$  is equal to the input  $x_m$ . The decoder process is described as follows:

$$y_m = g_{\theta^T}(h) = \sigma(W^T h + d), \quad (3)$$

where  $g_{\theta^T}$  is the decoder function,  $\sigma$  is the activation function of the decoder process,  $W^T$  represents the weight matrix between the hidden layer and the output layer of the network, and  $d$  is the bias vector generated by the decoder process.

The essence of AE training process is to optimize the network parameters  $\theta$  and  $\theta^T$ . In order to make the output  $y_m$  as close as possible to the input  $x_m$ , we characterize the degree of approximation between input and output by minimizing the reconstruction error  $J_{(\theta, \theta^T)}(x, y; W, b)$ . The optimization process is described below:

$$J_{(\theta, \theta^T)}(x, y; W, b) = \frac{1}{m} \|y - x\|^2. \quad (4)$$

In each training process, the gradient descent method is used to update the training parameters  $\theta$  and  $\theta^T$  of the AE network. The processes of network parameter update are as follows:

$$W_l = W_l - \alpha \frac{\partial}{\partial W_l} J_{(\theta, \theta^T)}(x, y; W, b), \quad l = 1, 2, \quad (5)$$

$$b_l = b_l - \alpha \frac{\partial}{\partial b_l} J_{(\theta, \theta^T)}(x, y; W, b), \quad l = 1, 2,$$

where  $\alpha$  represents the learning rate and partial derivatives  $(\partial/\partial W_l)J_{(\theta, \theta^T)}(x, y; W, b)$  and  $(\partial/\partial b_l)J_{(\theta, \theta^T)}(x, y; W, b)$  can be calculated with back propagation algorithm.

DNN can be simply viewed as a multihidden layers neural network formed by stacking many AutoEncoders. This model uses the bottom-up method of unsupervised learning, extracting the features layer by layer. Then supervised learning method is applied to fine-tune the whole network parameters, which can extract the most essential characteristics from original signals. The structure of DNN is shown in Figure 2.

First of all, pretrain the DNN by using the unsupervised layer-by-layer greedy training algorithm. Firstly, the first AutoEncoder AE1 is trained by giving an unlabeled dataset  $x$  as the input of encoder network. The encoded feature  $h_1$  is the hidden layer of AE1. The training parameter  $\theta_1$  is obtained by designing the unique  $x$  as the output of AE1. Then, use  $h_1$  as the input of the second AutoEncoder (AE2) and train AE2 to acquire the network training parameter  $\theta_2$ .  $h_2$  is the hidden layer of AE2 which can be viewed as the characteristics of AE2. After that, choose  $h_2$  as the input of the third AutoEncoder (AE3). Repeat the process to get the hidden layer features  $h_N$  of the  $N$ th AutoEncoder (AE  $N$ ) and the corresponding network training parameter  $\theta_N$ .

Secondly, a classifier is added in the top layer of DNN. The feature information is extracted by using the unsupervised learning method in the pretraining process of DNN. However, DNN does not have the ability of classifying; a classifier should be added in the top of DNN. In this paper, Softmax

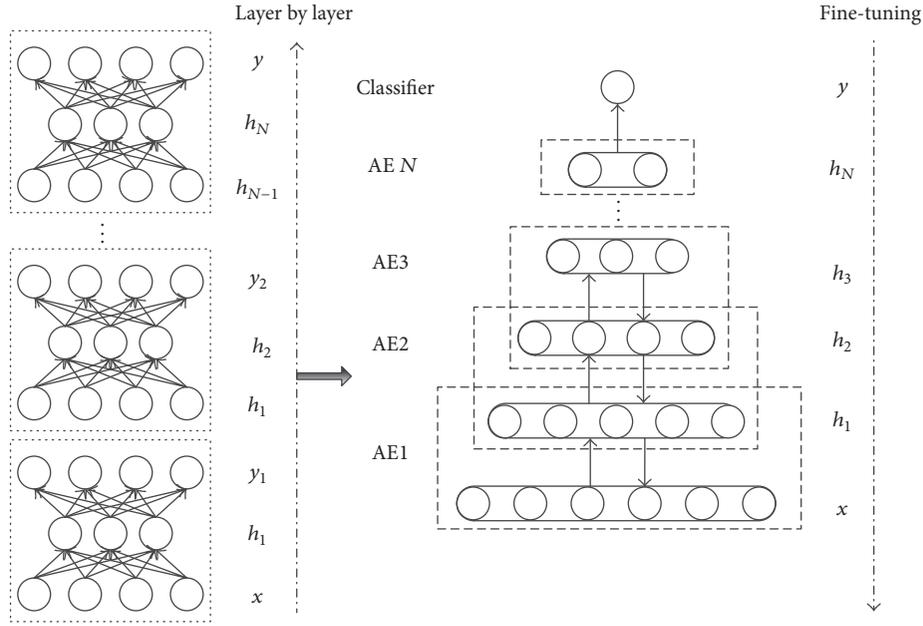


FIGURE 2: The structure of DNN.

classifier is used as the output layer of DNN. We suppose the training dataset is  $\{x_m\}$  ( $m = 1, 2, \dots, M$ ), the label is  $u_m \in \{1, 2, \dots, k\}$ , and the probability  $p(u = i | x)$  for each category  $i$  ( $i = 1, 2, \dots, k$ ) can be calculated via the following hypothesis function:

$$h_{\theta}(x_m) = \begin{bmatrix} p(u_m = 1 | x_m; \theta) \\ p(u_m = 2 | x_m; \theta) \\ \vdots \\ p(u_m = k | x_m; \theta) \end{bmatrix} = \frac{1}{\sum_{i=1}^k e^{\theta_i^T x_m}} \begin{bmatrix} e^{\theta_1^T x_m} \\ e^{\theta_2^T x_m} \\ \vdots \\ e^{\theta_k^T x_m} \end{bmatrix}, \quad (6)$$

where  $\theta$  is the model parameter of Softmax. Similarly to the AE model, in order to guarantee the performance of the classifier, the classifier model parameter is trained by minimizing the cost function  $J_{\theta}$ . The cost function of Softmax training process is shown in (7), where the top network parameter  $\theta_{N+1}$  is obtained from minimizing  $J_{\theta}(x_m)$ .

$$J_{\theta}(x_m) = -\frac{1}{M} \left[ \sum_{m=1}^M \sum_{i=1}^k 1\{u_m = i\} \log \frac{e^{\theta_i^T x_m}}{\sum_{i=1}^k e^{\theta_i^T x_m}} \right]. \quad (7)$$

Finally, fine-tune. In order to guarantee the accuracy of feature extraction and the classification effectiveness of output layer, the whole DNN training parameters are fine-tuned by using a supervise algorithm of back propagation

with some limited number of sample labels. The process of fine-tuning is completed by minimizing the reconstruction error  $E(\theta)$ . The procedures for parameter update are as follows:

$$E(\theta) = \frac{1}{M} \sum J_{\theta}(Y'_m, u_m; \theta), \quad (8)$$

$$\theta = \theta - \alpha \frac{\partial E(\theta)}{\partial \theta},$$

where  $Y'_m$  represents the actual output value,  $\theta$  is a parameter set generated from the whole network training,  $\theta = \{\theta_1, \theta_2, \dots, \theta_N, \theta_{N+1}\}$ , back propagation algorithm is used to update the network parameter  $\theta$ , and  $\alpha$  is the learning rate in the process of deep learning. The fine-tuning process uses the labeled data to improve the performance of DNN.

**2.2. DNN-Based Classification.** In order to accurately extract the essential characteristics of the mechanical equipment health conditions by DNN modeling, the following steps are required. Firstly, the original vibration signals should be preprocessed. Since frequency-domain signals are more sensitive to mechanical equipment faults, the original time-domain signals are converted into frequency-domain signals in the first step. Secondly, use the preprocessed data as the input of the DNN model to extract features of mechanical equipment health conditions with unsupervised layer-by-layer pretraining. Last but not least, the whole network parameter  $\theta$  can be updated by using the back propagation algorithm to fine-tune the DNN structure when limited number of labeled samples is available. In this way we can get an effective feature extraction result for fault classification. The preprocessed datasets are divided into training data and testing data. The training data is used to construct DNN model to obtain the training parameter  $\theta$ , and the

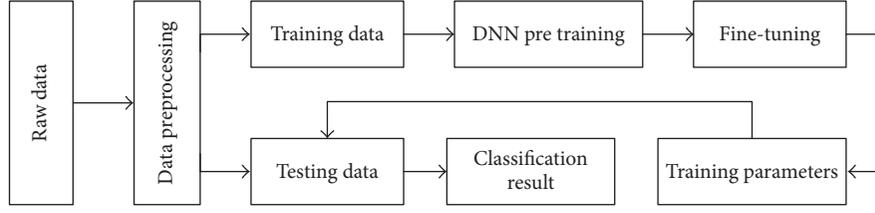


FIGURE 3: Framework of fault classification based on DNN.

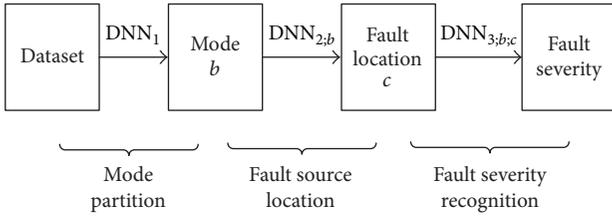


FIGURE 4: A hierarchical DNN framework of three layers.

testing network initialized with training parameter  $\theta$  is used to verify its effectiveness. Misclassification rate is used as an accuracy indicator of the DNN-based fault classification method. Detailed steps of DNN for mechanical system fault classification are shown in Figure 3.

### 3. Multimode Fault Classification Model Based on Deep Learning

There are a number of multimode processes in practical system. For multimode process, the potential feature extracted from the observation of each steady mode also varies. So it is necessary to separate the observation into several operation modes for accuracy data feature extraction.

Therefore, mode partition is a fundamental step before fault classification. In this paper, this problem is solved by constructing a hierarchical DNN model with the first hierarchy specially devised for the purpose of mode partition. By this means, it can make an effective mode partition for multimode process, which can increase the accuracy of DNN-based fault classification. Framework of three-layer DNN is shown in Figure 4.

The detailed steps for multimode fault classification are as follows.

*Step 1* (mode partition). In this step, we focus on building a DNN model to determine the mode label of each sample. The whole datasets are used as the input of the multimode classification model. The mode partition process can be illustrated in detail as follows.

(1) Construct a new  $DNN_1$  with  $N$  hidden layers AE described in (9), and initialize the training parameters of  $DNN_1$ .

$$\begin{aligned} & [\text{Net}_1, \text{Tr}_1] \\ & = \text{Feedforward}(\theta'_{11}; H_{11}, H_{12}, \dots, H_{1N}; S_1), \end{aligned} \quad (9)$$

where  $\theta'_{11} = \{W_1, b_1\}$ , where  $W_1$  is the weight matrix and  $b_1$  is the bias vector.  $H_{11}, H_{12}, \dots, H_{1N}$  are the numbers of hidden layer neurons in  $DNN_1$ . The network configuration can be represented by  $\text{Tr}_1$ .  $S_1$  denotes the training dataset. We use  $M_{11}$  in (10) to represent the number of neurons in the input layer of  $DNN_1$ .

$$M_{11} = \text{size}(S_1). \quad (10)$$

The parameters of  $DNN_1$  can be initialized via

$$\begin{aligned} W_1 &= \text{rand}(H_{11}, M_{11}), \\ b_1 &= \text{zeros}(H_{11}, 1). \end{aligned} \quad (11)$$

(2) Training of  $DNN_1$  to obtain the net parameter  $\theta'_1$ . Unsupervised layer-by-layer feature extraction based on the training dataset  $S_1$  is implemented to the  $N$ -level AE defined in (9).

$$\begin{aligned} h_1 &= f_{\theta'_{11}}(S_1) = \sigma(W_1 \cdot S_1 + b_1), \\ h_N &= f_{\theta'_{1N}}(h_{N-1}). \end{aligned} \quad (12)$$

Add a Softmax classifier on the top of  $DNN_1$ . Limited number of training labels sets  $\rho_1$  is used to fine-tune  $DNN_1$  and update the training parameter  $\theta'_1$  via

$$\begin{aligned} E(\theta'_1) &= \min \frac{1}{m} \sum J_{\theta'_1}(Y_1, \rho_1; \theta'_1), \\ \theta'_1 &= \theta'_1 - \alpha_1 \frac{\partial E(\theta'_1)}{\partial \theta'_1}, \end{aligned} \quad (13)$$

where  $\theta'_1 = \{\theta'_{11}, \theta'_{12}, \dots, \theta'_{1N}, \theta'_{1(N+1)}\}$ , with  $\theta'_{1(N+1)}$  calculated by (6)-(7), and  $m$  is the number of samples.  $Y_1$  denotes the output of  $DNN_1$ , and  $\alpha_1$  is learning rate in fine-tuning process.

(3) Mode partition uses the trained  $DNN_1$ . Once test sample  $S'_1$  is obtained, compute the probability of each test sample via the trained  $\text{Net}_1$ . Then use (14) to divide the test sample into different modes:

$$\begin{aligned} & \text{Mode}(m) \\ & = \arg \max_b P(Y_1(m) = b | S'_1(m); \theta'_1; \text{Tr}_1), \end{aligned} \quad (14)$$

where  $m = 1, 2, \dots, M$  and  $b$  ( $b = 1, 2, \dots, B$ ) is the mode type of sample.  $\text{Mode}(m)$  denotes the mode label of the  $m$ th test sample.

Compare the mode partition label  $\text{Mode}(m)$  with the actual mode label  $\text{Label}(m)$  to determine the misclassification number as

$$e_1 = \text{size}(S_{\text{miss}}), \quad (15)$$

where  $\text{size}$  is the operation to characterize the size of a set and  $S_{\text{miss}}$  is the misclassification set defined by

$$S_{\text{miss}} = \left\{ x_m \mid \arg \underset{x_m}{\text{Mode}}(m) \neq \text{Label}(m) \right\}. \quad (16)$$

*Step 2* (fault source location). For a certain mode partitioned in Step 1, We can further locate the fault source. The procedure in Step 2 is analogous to Step 1, which is described below.

(1) According to the mode partition result, we build the second hierarchy of the model which comprises a set of  $B$  DNNs, and  $S_{2,b}$  ( $b = 1, 2, \dots, B$ ) denotes the training dataset in  $\text{DNN}_2$ .

$$\begin{aligned} & [\text{Net}_{2,b}, \text{Tr}_{2,b}] \\ &= \text{Feedforward}(\theta'_{2,b1}; H_{2,b1}, H_{2,b2}, \dots, H_{2,bN}; S_{2,b}). \end{aligned} \quad (17)$$

Parameter initialization mechanism of  $\text{DNN}_2$  is the same as Step 1.

(2) Train  $\text{DNN}_2$  to obtain the net parameters  $\theta'_{2,b}$ . Similarly, for detailed calculation process, one can refer to (12)–(13).

(3) Determine the fault location by the trained  $\text{Net}_{2,b}$ .

The test dataset  $S'_{2,b}$  is used to predict the unknown fault locations based trained  $\text{Net}_{2,b}$ . Assume that each mode has  $c$  ( $c = 1, 2, \dots, C$ ) different fault locations; fault location label for the  $m$ th sample of the  $b$ th mode can be calculated with prediction formula via

$$\begin{aligned} & \text{Location}_{2,b}(m) \\ &= \arg \max_c P(Y_{2,b}(m) = c \mid S'_{2,b}(m); \theta'_{2,b}; \text{Tr}_{2,b}). \end{aligned} \quad (18)$$

Compute the misclassification number  $e_{2,b}$  ( $b = 1, 2, \dots, B$ ) of the  $b$ th mode. And then the misclassification of this classification step can be computed via

$$e_2 = \sum_{b=1}^B e_{2,b}. \quad (19)$$

*Step 3* (fault severity recognition). In order to identify the fault severity, the third hierarchy is devised with the intention to distinguish the fault severity. Construct the third deep network  $\text{Net}_{3,b;c}$ ,  $S_{3;b;c}$  is the training dataset in  $\text{DNN}_3$ , and  $S'_{3;b;c}$  is the test dataset. Parameter training process is similar

to Step 2. The severity classification label of the  $m$ th sample in  $S'_{3;b;c}$  can be determined by

$$\begin{aligned} & [\text{Net}_{3;b;c}, \text{Tr}_{3;b;c}] = \text{Feedforward}(\theta'_{3;b;c1}; H_{3;b;c1}, H_{3;b;c2}, \\ & \dots, H_{3;b;cN}; S_{3;b;c}) \end{aligned} \quad (20)$$

$$\begin{aligned} & \text{Severity}_{3;b;c}(m) \\ &= \arg \max_d P(Y_3(m) = d \mid S'_{3;b;c}(m); \theta'_{3;b;c}; \text{Tr}_{3;b;c}). \end{aligned} \quad (21)$$

The misclassification number for a given fault in a certain mode can be computed via

$$\begin{aligned} e_{3;c} &= \sum_{b=1}^B e_{3;b;c}, \\ e_3 &= \sum_{c=1}^{BC} e_{3;c}, \end{aligned} \quad (22)$$

where  $e_{3;b;c}$  is the misclassification number of the  $c$ th fault location in the  $b$ th mode,  $e_{3;c}$  is the misclassification number of all  $C$  modes, and  $e_3$  is the misclassification number in this step.

*Step 4* (accuracy computation of the whole multimode classification network). In this paper, the classification accuracy of the hierarchical DNN is measured by the numbers of misclassifications. The final accuracy is calculated by the ratio of the total number of the misclassifications to the total number of samples. The procedure of calculation is as follows:

$$\text{Correct\_rate} = \left(1 - \frac{e_3}{M}\right) \times 100\%. \quad (23)$$

Combining (21) with (22), the final accuracy of the proposed multimode fault classification based on DNN can be formulated as

$$\text{Correct\_rate} = \left(1 - \frac{\sum_{b=1}^B \sum_{c=1}^C e_{3;b;c}}{M}\right) \times 100\%, \quad (24)$$

where  $N$  is the number of total samples, and the flow chart of the proposed multimode fault classification method based on three-layer DNN is depicted in Figure 5.

#### 4. Application to Rolling Bearing Fault Classification

Rolling bearings play an important role for rotating machinery. The health condition of the bearing directly affects the reliability and stability in the whole system. Rolling bearing as the experimental platform is used to verify the effectiveness of the hierarchical DNN multimode fault classification method, and the performance of the proposed method is compared with the traditional method such as DNN, BPNN, SVM, hierarchical BPNN, and hierarchical SVM, which is listed in detail in Section 4.3.

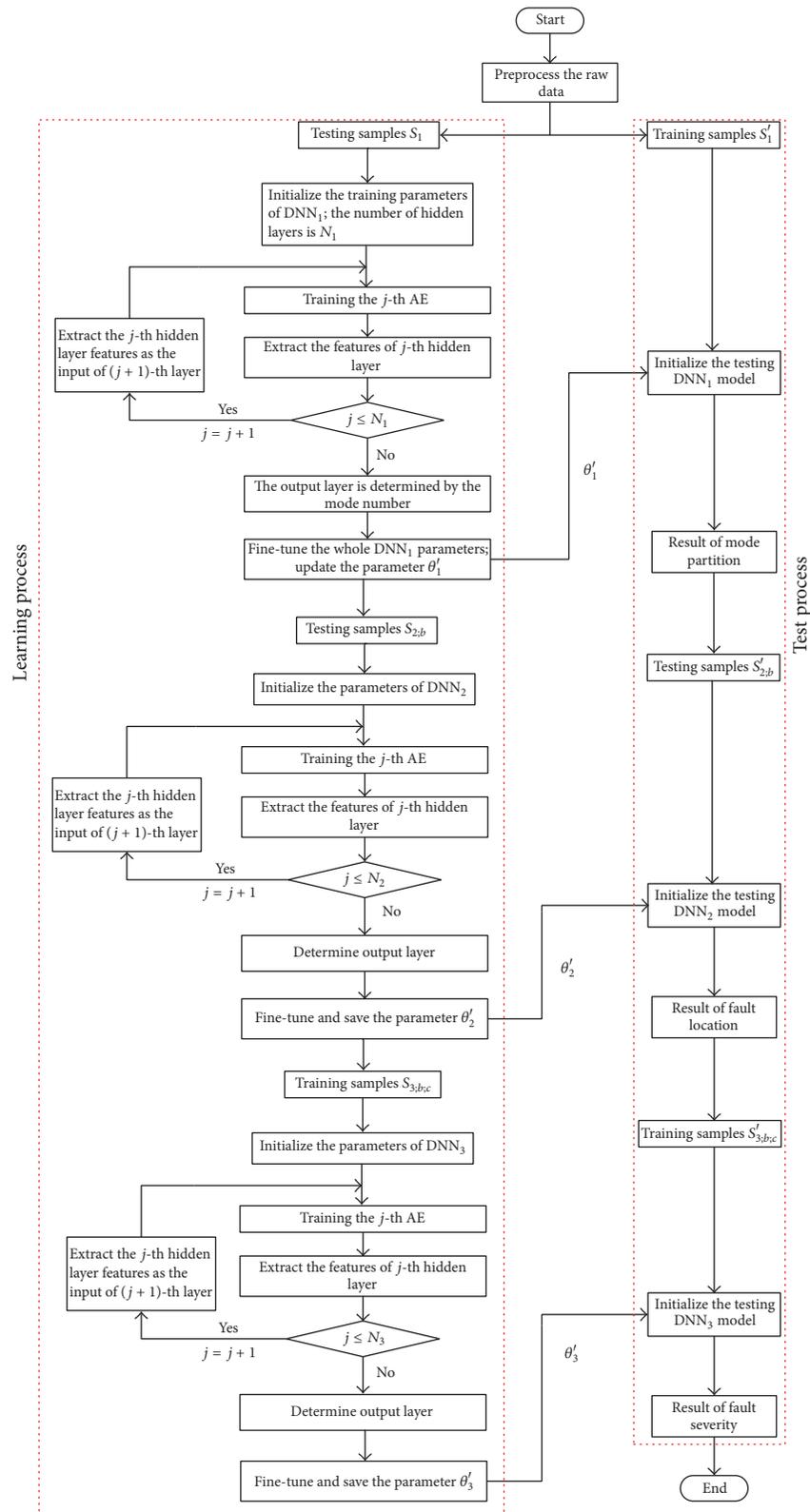


FIGURE 5: Flow chart of multimode classification based on DNN.

TABLE 1: Four modes of rolling bearing.

Mode	Load (hp)	Rotating speed (rpm)
Mode 1	0	1797
Mode 2	1	1772
Mode 3	2	1750
Mode 4	3	1730

TABLE 2: Data description of in dataset for a given mode.

Dataset for different fault location	Dataset for different fault severity	Sample number More/fewer	Fault type	Fault size /mm
		200/100	Normal (N)	0.000
$S_{b;1}$	$S_{b;1;1}$	200/100	Inner race fault (IF)	0.007
	$S_{b;1;2}$	200/100	Inner race fault (IF)	0.014
	$S_{b;1;3}$	200/100	Inner race fault (IF)	0.021
$S_{b;2}$	$S_{b;2;1}$	200/100	Outer race fault (OF)	0.007
	$S_{b;2;2}$	200/100	Outer race fault (OF)	0.014
	$S_{b;2;3}$	200/100	Outer race fault (OF)	0.021
$S_{b;3}$	$S_{b;3;1}$	200/100	Roller fault (RF)	0.007
	$S_{b;3;2}$	200/100	Roller fault (RF)	0.014
	$S_{b;3;3}$	200/100	Roller fault (RF)	0.021

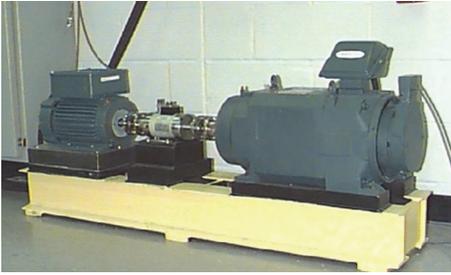


FIGURE 6: Experimental platform for acquiring the vibration signals of rolling bearing.

**4.1. Experimental Platform.** The experimental datasets are obtained from the Case Western Reserve University Bearing Data Center in the United States [35]. The experimental platform is shown in Figure 6. It can be seen that the experimental platform consists of a 2 hp motor, a power meter, an electronic controller, a torque sensor, and a load motor. The vibration signals of the drive end of the motor are collected by the acceleration sensor as the experimental datasets for bearing fault classification. In this experiment, we use acceleration sensor to collect the vibration signals with the load of 0 hp, 1 hp, 2 hp, and 3 hp, respectively, and the sampling frequency is 48 kHz. There are four types of bearing health condition: (1) normal condition; (2) inner race fault; (3) outer race fault; (4) roller fault. The sizes of the bearing fault were 0.007 mm, 0.014 mm, and 0.021 mm, respectively.

**4.2. Data Description.** In this case, we collect the vibration signals of the bearing drive end at different loading. The dataset collected contains 4 kinds of modes; the motor load is 0 hp, 1 hp, 2 hp, and 3 hp, respectively, and 4 modes are shown in Table 1. In each mode, there are four states of inner race fault, outer race fault, roller fault, and normal, with 3 different fault sizes in each fault state, that is to say 10 different fault types in a single mode. This paper selects 200 samples in each fault type; each sample contains 2048 observation points. 100 samples are randomly selected as the training data, and the other 100 samples as the testing data. We use Fast Fourier Transform (FFT) for each sample to get 2048 Fourier coefficients. Because of the symmetry of the Fourier coefficients, we take the first 1024 coefficients as the new samples; that is to say the dataset contains 8000 samples. In order to compare the proposed method of hierarchical network with single-layer network and explore the effect of different sample numbers on network, for a given mode, the sample number of each DNN is listed in Table 2. In addition, we present the original time-domain waveforms of the 10 fault types in mode 1 under  $A$ , as shown in Figure 7.

**4.3. Results of Fault Classification.** The proposed hierarchical DNN structure is applied to bearing fault classification; there are 8000 samples, 4 different modes, 4 fault positions in each mode, and totally 40 health conditions in dataset  $A$ . The health conditions of rotating machinery system under multimode, multicondition, multifault type, and large sample data are simulated which demonstrated the performance with

TABLE 3: DNN model parameters.

Training parameter	DNN <sub>1</sub>	DNN <sub>2;b</sub>	DNN <sub>3;b;c</sub>
Hidden layers	5	4	3
Number of neurons	512/400/300/200/100	512/400/200/100	512/256/100
Max number of epochs	500	300	300

TABLE 4: Accuracy of classification in time domain and frequency domain.

Method	Time domain data	Frequency domain data
HDNN	80.65	99.96

TABLE 5: Fault severity classification result comparison after mode partition.

Method	Accuracy of fault classification	Accuracy of fault severity recognition
HDNN	99.79	99.52
DNN	97.06	96.38
HSVM	82.82	77.00
SVM	65.74	58.40
HBPNN	81.28	71.68
BPNN	68.11	62.42

the proposed method. To reduce the effect of randomness, the experiment was repeated 20 times. In this paper, the initialized parameters in the DNN pretraining process are shown in Table 3.

The network training uses stochastic gradient descent method; on each hierarchy the maximum number of iterations of DNN is 500, 300, and 300, respectively. Simulation of three traditional methods, BPNN, SVM, and DNN, is compared with simulation of the proposed multimode fault classification approach to verify its effectiveness. In addition, hierarchical BPNN (HBPNN) and hierarchical SVM (HSVM) are also compared with hierarchical DNN (HDNN). BPNN uses the gradient descent method to update the network weights and bias parameter; one-to-one training mechanism is used to train a SVM with radial basis. The training mechanism of HBPNN and HSVM is the same as HDNN.

Table 4 compares the fault classification accuracies in time domain and frequency domain. It can be seen from Table 4 that rotation machinery fault is more sensitive in frequency domain. So we use FFT as a tool to preprocess the original data.

Table 5 compares the fault classification results after mode partition. It can be seen from line 2 and line 3 that HDNN can obtain more accurate classification either for fault source location or for fault severity recognition which tells us that mode partition is a critical step in multimode fault classification.

The hierarchical model for the case of BPNN and SVM also confirms this conclusion. Comparing line 2

TABLE 6: The classification results by the second hierarchical of the proposed model.

Method	Mode partition
HDNN	99.96
HBPNN	90.45
HSVM	89.73

with line 4 and line 6, we can see that HDNN is significantly superior to other hierarchical machine learning models because of the fact that HDNN can get better mode partition accuracy which is shown in Table 6. On the other hand, we can draw another conclusion that the performance of traditional BPNN method is superior to the traditional SVM method in the large sample case, but the accuracy of HSVM is higher than that of HBPNN due to the fact that SVM does well in small sample learning.

In order to demonstrate the performance of the proposed multimode classification method, the hierarchical machine learning methods are employed in this paper. As can be seen from Table 6, the accuracy of mode partition with proposed HDNN method can reach 99.96%, and we can naturally find that the performance of HDNN is superior to HBPNN and HSVM in mode partition procedure.

In view of the excellent performance of the proposed multimode classification method, we found that the performance

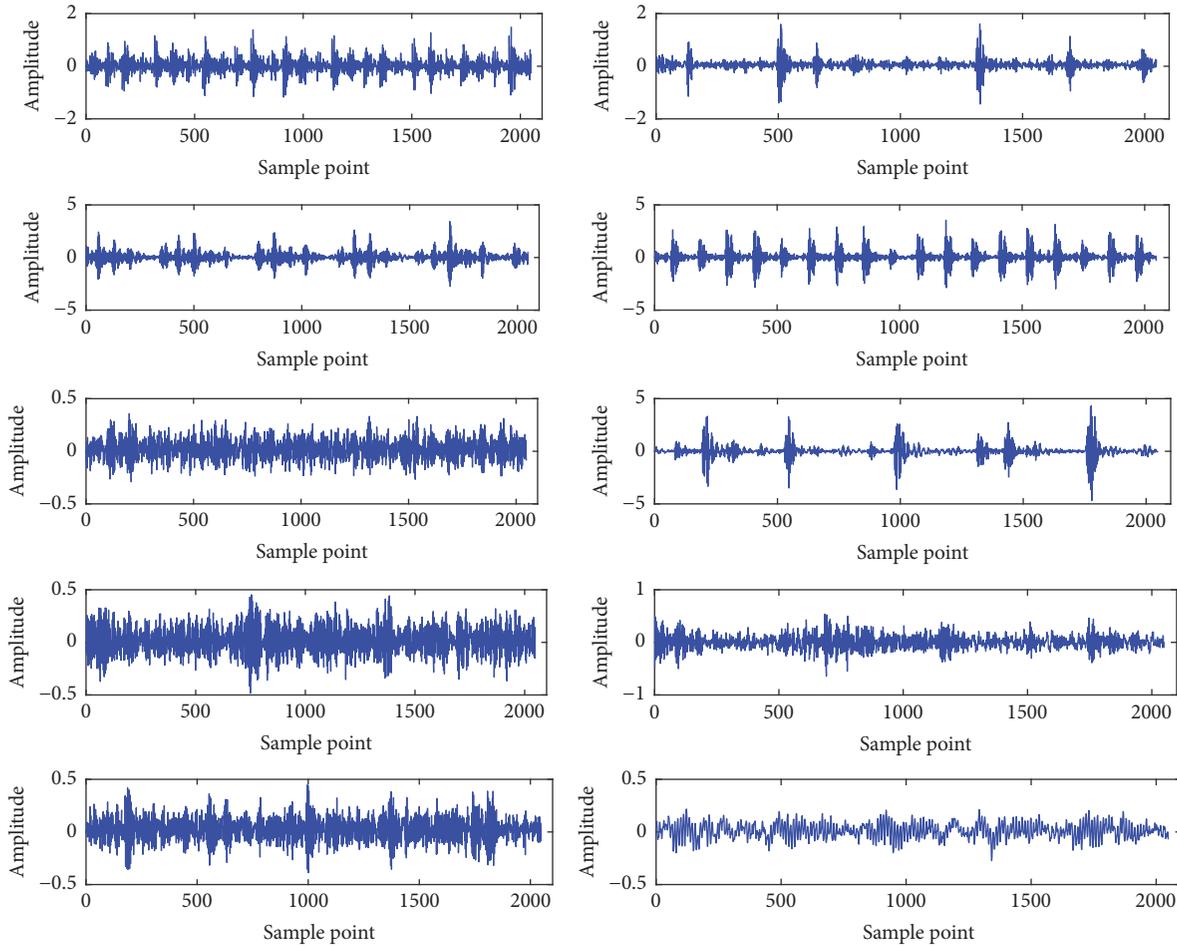


FIGURE 7: Observation of original signals corresponding to 10 fault types.

of classification was influenced by accurate feature extraction. In order to verify the effectiveness of HDNN based feature extraction method, scatter plots of the feature extracted are demonstrated in Figures 8–10. As shown in Table 3, in each training process, the number of neurons in the last hidden layer is 100; that is to say, the feature dimension is 100, which is too large to be visualized. Therefore, PCA is used as a data compression tool to reduce the feature dimension. In this paper, we use the first three key principal components to plot the scatter chart of the fault source location feature extracted by HDNN, as shown in Figure 8. Figure 8 is the scatter plots for fault feature extracted by HDNN after mode partition, while Figure 9 shows the scatter plots for fault feature extracted by DNN without mode partition. From Figures 9 and 10, we can see that some fault features are overlapped, which result in an unsatisfactory fault classification result.

Figure 10 is the scatter plot of the feature extracted for different modes. We can see from Figure 10 that HDNN does well in multimode fault feature extraction which will greatly affect the accuracy of the successive fault classification.

In summary, the proposed multimode classification method can accurately extract the different fault features based on its strong nonlinear characterization ability.

In general, efficiency of the fault classification method is affected by sample number of the train data. Figure 11 displays the fault classification accuracy of DNN and HDNN in two cases. Red line denotes the classification accuracy of the case when more samples are used as the training data. Black line denotes the classification accuracy of the case when fewer samples (only 1/2 of the first case) are used as the training data. In addition, the line with “\*” is the simulation result of HDNN and the line with “□” is the simulation result of traditional DNN.

From Figure 11, it can be clearly seen that (1) fault classification accuracy of HDNN does not vary much for the two cases, while the fault classification accuracy of DNN is greatly affected by the number of training data used and (2) in both cases fault classification accuracy of HDNN is much better than DNN. So we can come to the conclusion that HDNN is a more robust fault classification for multimode bearing fault classification in the case when fewer number of training data are available.

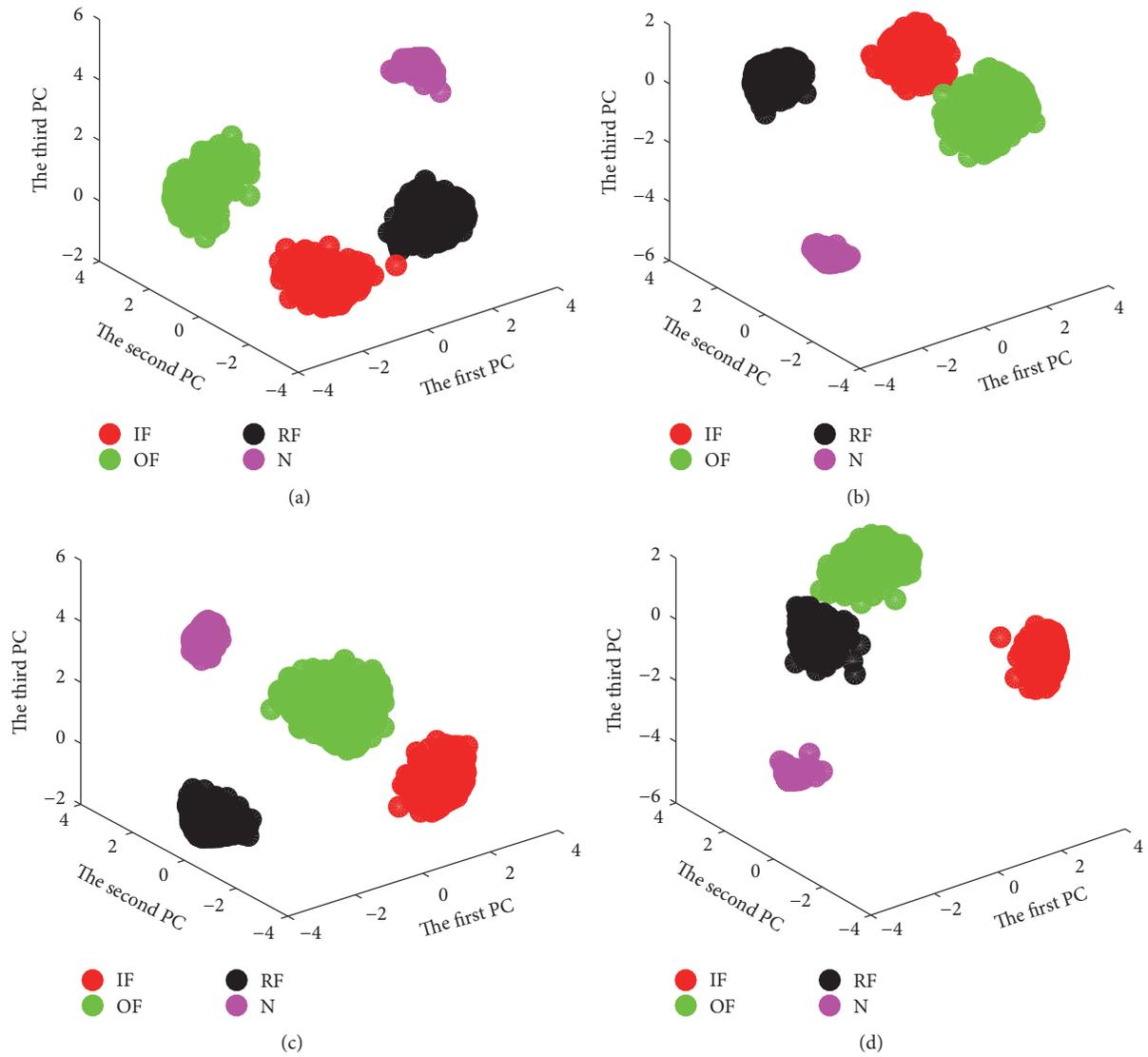


FIGURE 8: Scatter plots of principal components for the feature of fault classification; (a)-(d) represent four modes: corresponding to Mode 1, Mode 2, Mode 3, and Mode 4, respectively.

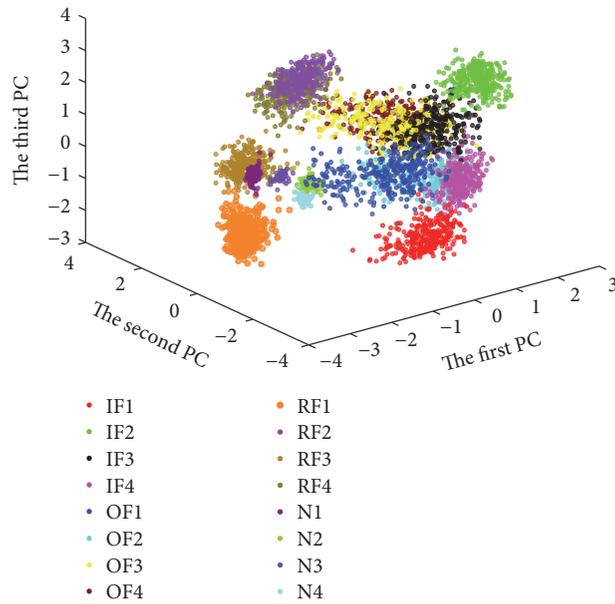


FIGURE 9: Scatter plots of principal components for fault features with traditional DNN method.

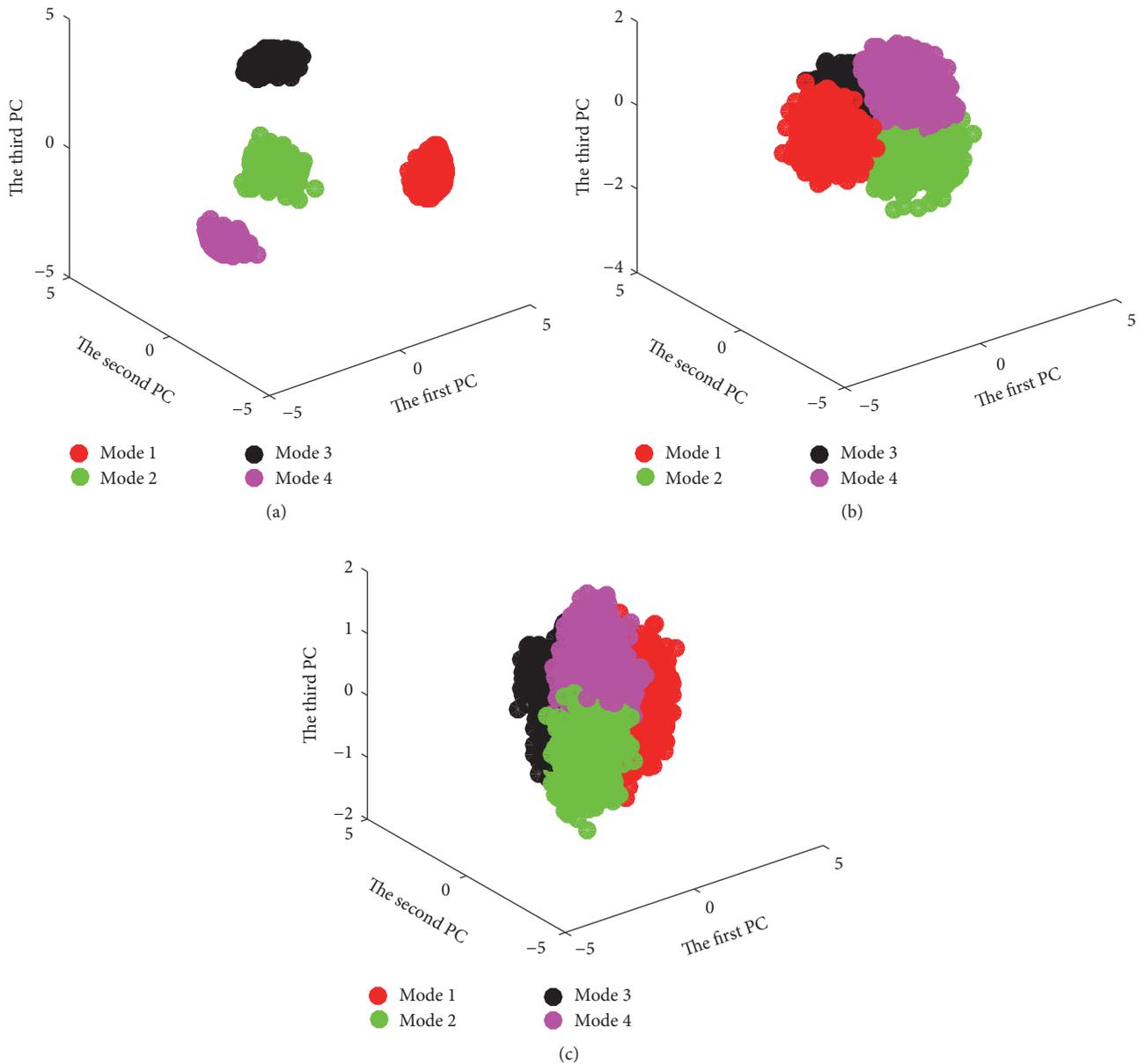


FIGURE 10: Scatter plots of principal components for mode partition features; (a)–(c) represent mode partition result corresponding to HDNN, HBPNN, and HSVM, respectively.

## 5. Conclusions

In this paper, a novel multimode fault classification method based on DNN is developed. The main idea is to construct a hierarchical DNN model with the first hierarchy specially devised for the purpose of mode partition. The second hierarchical model comprising a set of DNNs is devised to extract feature separately of different modes and precisely diagnose the fault source. Another set of DNNs is devised to distinguish the severity of a certain fault in a given mode, which is helpful for predictive maintenance of the machinery equipment. Rolling bearing is the

experiment platform to verify the efficiency of the proposed method.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was supported in part by the Natural Science Fund of China (Grant nos. U1604158, U1509203, and

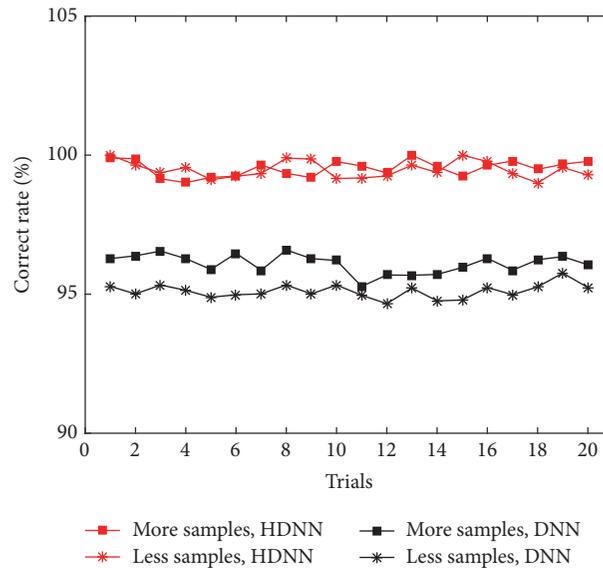


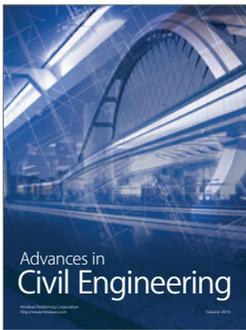
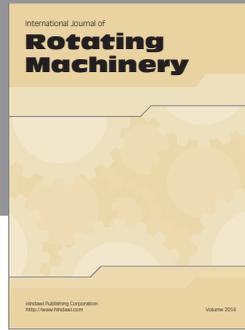
FIGURE 11: Robustness of the fault classification method to the sample number of training data with 20 trials.

61333005) and Technical Innovation Talents Scheme of Henan Province (Grant no. 2012HASTIT005).

## References

- [1] D.-H. Zhou, Y. Liu, and X. He, "Review on fault diagnosis techniques for closed-loop systems," *Acta Automatica Sinica*, vol. 39, no. 11, pp. 1933–1943, 2013.
- [2] F. N. Zhou, J. H. Park, and Y. J. Liu, "Differential feature based hierarchical PCA fault detection method for dynamic fault," *Neurocomputing*, vol. 202, pp. 27–35, 2016.
- [3] C. Li, R.-V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, and R. E. Vásquez, "Multimodal deep support vector classification with homologous features and its application to gearbox fault diagnosis," *Neurocomputing*, vol. 168, pp. 119–127, 2015.
- [4] X. He, Z. Wang, Y. Liu, and D. H. Zhou, "Least-squares fault detection and diagnosis for networked sensing systems using a direct state estimation approach," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1670–1679, 2013.
- [5] D. Zhao, D. Shen, and Y. Q. Wang, "Fault diagnosis and compensation for two-dimensional discrete time systems with sensor faults and time-varying delays," *International Journal of Robust and Nonlinear Control*, 2017.
- [6] H. Li and D. Y. Xiao, "Surver on data driven fault classification methods," *Control and Decision*, vol. 26, no. 1, pp. 1–16, 2011.
- [7] R. M. An and Y. Gao, "Spacecraft fault classification based on hierarchical neural network," *Spacecraft Environment Engineering*, vol. 30, no. 2, pp. 203–208, 2013.
- [8] F. N. Zhou, C. L. Wen, Y. B. Leng, and Z. G. Chen, "A data-driven fault propagation analysis method," *Journal of Chemical Industry and Engineering (China)*, vol. 61, no. 8, pp. 1993–2001, 2010.
- [9] H. Ji, X. He, and D. Zhou, "On the use of reconstruction-based contribution for fault diagnosis," *Journal of Process Control*, vol. 40, pp. 24–34, 2016.
- [10] D. J. Yu, M. F. Chen, J. S. Cheng, and Y. Yang, "A fault classification approach for rotor systems based on empirical mode decomposition method and support vector machines," *Proceedings of the Chinese Society for Electrical Engineering*, vol. 26, no. 16, pp. 162–167, 2006.
- [11] M. Gan, C. Wang, and C. A. Zhu, "Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings," *Mechanical Systems and Signal Processing*, vol. 72-73, pp. 92–104, 2016.
- [12] G. F. Bin, J. J. Gao, X. J. Li, and B. S. Dhillon, "Early fault diagnosis of rotating machinery based on wavelet packets—empirical mode decomposition feature extraction and neural network," *Mechanical Systems and Signal Processing*, vol. 27, no. 1, pp. 696–711, 2012.
- [13] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 21, no. 6, pp. 2560–2574, 2007.
- [14] D. Zhao, Z. Lin, and Y. Wang, "Integrated state/disturbance observers for two-dimensional linear systems," *IET Control Theory & Applications*, vol. 9, no. 9, pp. 1373–1383, 2015.
- [15] Q. Hu, Z. He, Z. Zhang, and Y. Zi, "Fault diagnosis of rotating machinery based on improved wavelet package transform and SVMs ensemble," *Mechanical Systems and Signal Processing*, vol. 21, no. 2, pp. 688–705, 2007.
- [16] L. Y. Wang, W. G. Zhao, and Y. Liu, "Rolling bearing fault diagnosis based on wavelet packet- neural network characteristic entropy," *Advanced Materials Research*, vol. 108–111, no. 1, pp. 1075–1079, 2010.
- [17] Y. Yang and W. Tang, "Study of remote bearing fault diagnosis based on BP Neural Network combination," in *Proceedings of the 7th International Conference on Natural Computation (ICNC '11)*, pp. 618–621, IEEE, Shanghai, China, July 2011.
- [18] L. Jiang, Q. Li, J. Cui, and J. Xi, "Rolling bearing fault diagnosis based on higher-order cumulants and BP neural network," in *Proceedings of the 27th Chinese Control and Decision Conference (CCDC '15)*, pp. 2664–2667, IEEE, Qingdao, China, May 2015.
- [19] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time series forecasting using a deep belief network with restricted

- Boltzmann machines,” *Neurocomputing*, vol. 137, pp. 47–56, 2014.
- [20] S. M. Zhang, F. L. Wang, S. Tan, and S. Wang, “A fully automatic online mode identification method for multi-mode processes,” *Acta Automatica Sinica*, vol. 42, no. 1, pp. 60–80, 2016.
- [21] J. Schmidhuber, “Deep Learning in neural networks: an overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [22] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [23] H. Ze, A. Senior, and M. Schuster, “Statistical parametric speech synthesis using deep neural networks,” in *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '13)*, pp. 7962–7966, IEEE, Vancouver, Canada, May 2013.
- [24] B. Song, S. Tan, and H. B. Shi, “Key principal components with recursive local outlier factor for multimode chemical process monitoring,” *Journal of Process Control*, vol. 47, pp. 136–149, 2016.
- [25] L. Zhao, C. Zhao, and F. Gao, “Inter-batch-evolution-traced process monitoring based on inter-batch mode division for multiphase batch processes,” *Chemometrics and Intelligent Laboratory Systems*, vol. 138, pp. 178–192, 2014.
- [26] Y. Zhang, C. Wang, and R. Lu, “Modeling and monitoring of multimode process based on subspace separation,” *Chemical Engineering Research and Design*, vol. 91, no. 5, pp. 831–842, 2013.
- [27] F.-N. Zhou, C.-L. Wen, T.-H. Tang, and Z.-G. Chen, “DCA based multiple faults diagnosis method,” *Acta Automatica Sinica*, vol. 35, no. 7, pp. 971–982, 2009.
- [28] P. Tamilselvan and P. Wang, “Failure diagnosis using deep belief learning based health state classification,” *Reliability Engineering & System Safety*, vol. 115, pp. 124–135, 2013.
- [29] R. Huang, C. Liu, G. Li, and J. Zhou, “Adaptive deep supervised autoencoder based image reconstruction for face recognition,” *Mathematical Problems in Engineering*, vol. 2016, Article ID 6795352, 14 pages, 2016.
- [30] H. Liu, L. Li, and J. Ma, “Rolling bearing fault diagnosis based on STFT-deep learning and sound signals,” *Shock and Vibration*, vol. 2016, Article ID 6127479, 12 pages, 2016.
- [31] P. L. Wang and C. J. Xia, “Fault detection and self-learning identification based on PCA-PDBNs,” *Chinese Journal of Scientific Instrument*, vol. 36, no. 5, pp. 1147–1154, 2015.
- [32] R. Pang, Z. B. Yu, W. Y. Xiong, and H. Li, “Faults recognition of high-speed train bogie based on deep learning,” *Journal of Railway Science and Engineering*, vol. 12, no. 6, pp. 1283–1288, 2015.
- [33] C. Lu, Z. Y. Wang, W. L. Qin, and J. Ma, “Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification,” *Signal Processing*, vol. 130, pp. 377–388, 2017.
- [34] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, “Deep neural networks: a promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data,” *Mechanical Systems and Signal Processing*, vol. 72–73, pp. 303–315, 2016.
- [35] Bearing Data Centre, Case Western Reserve University, <http://csegroups.case.edu/bearingdatacenter/home>.



**Hindawi**

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

