Hindawi Publishing Corporation
Shock and Vibration
Volume 2016, Article ID 6127479, 12 pages
http://dx.doi.org/10.1155/2016/6127479

Research Article

Rolling Bearing Fault Diagnosis Based on STFT-Deep Learning and Sound Signals

Hongmei Liu,[1] Lianfeng Li,[1] and Jian Ma[1,2]

[1]*School of Reliability and Systems Engineering, Beihang University, Beijing, China*
[2]*Science & Technology on Reliability & Environmental Engineering Laboratory, Beijing, China*

Correspondence should be addressed to Jian Ma; 09977@buaa.edu.cn

Received 26 April 2016; Accepted 20 July 2016

Academic Editor: Fiorenzo A. Fazzolari

The main challenge of fault diagnosis lies in finding good fault features. A deep learning network has the ability to automatically learn good characteristics from input data in an unsupervised fashion, and its unique layer-wise pretraining and fine-tuning using the backpropagation strategy can solve the difficulties of training deep multilayer networks. Stacked sparse autoencoders or other deep architectures have shown excellent performance in speech recognition, face recognition, text classification, image recognition, and other application domains. Thus far, however, there have been very few research studies on deep learning in fault diagnosis. In this paper, a new rolling bearing fault diagnosis method that is based on short-time Fourier transform and stacked sparse autoencoder is first proposed; this method analyzes sound signals. After spectrograms are obtained by short-time Fourier transform, stacked sparse autoencoder is employed to automatically extract the fault features, and softmax regression is adopted as the method for classifying the fault modes. The proposed method, when applied to sound signals that are obtained from a rolling bearing test rig, is compared with empirical mode decomposition, Teager energy operator, and stacked sparse autoencoder when using vibration signals to verify the performance and effectiveness of the proposed method.

1. Introduction

As one of the most common components in rotating machinery, rolling bearings play a key role in maintaining the normal operation of entire machines. The faults of rolling bearings usually lead to a considerable decline in industrial productivity and can even cause enormous economic losses. To increase productivity and to reduce undesirable casualties, condition monitoring and fault diagnosis attract broad attention. In addition, the maintenance cost can be reduced, especially if the faults are identified before they become severe.

The features of sound signals can be used to detect faults in machines; for example, in the regular maintenance of a railway system, maintenance workers use clicking echoes to identify whether the train wheels are healthy. If the echo is dull, then a wheel could have internal cracks; otherwise, it is most likely normal. Similarly, experienced maintenance men in other engineering fields can judge whether a machine runs normally by recognizing the sound features. Sounds that are produced during running are characteristic of operating under healthy conditions and differ across fault modes [1, 2]. Similarly, the sound signals change gradually while the components in the rolling bearings develop faults, and different faults produce different sounds. Because of these changes, the health status can be determined. At the same time, based on the differences between the fault modes, the various faults can be classified.

For fault diagnosis, high identification accuracy depends on having effective feature representations. However, noises and complex structures in the observed signal increase the difficulty of extracting valid characteristics. For this reason, a large amount of work regarding feature extraction and selection in fault diagnosis has been performed using different types of signals and algorithms.

In most of the existing diagnosis literatures based on vibration signals, the researchers either applied WT (wavelet transformation) to acquire time-frequency information of the signal and then extract features from the time-frequency spectra or employed EMD (empirical mode decomposition) [3], LMD (local mean decomposition) [4], and LCD (local

characteristic scale decomposition) [5] to adaptively decompose the original signal into a series of scales and then extract the energy or entropy, a complexity measure of the signal, to be the fault features. Usually, to cover sufficient fault information, it is inevitable that the dimension of the acquired features is sufficiently high so that visualization is difficult, while the classification performance can become poor. Therefore, a dimensionality reduction method, such as common PCA (principal component analysis) [6], KPCA (kernel principal component analysis) [7], ISOMAP (isometric feature mapping) [8], LLE (locally linear embedding) [9], or LTSA (local tangent space alignment) [10], is necessary to map high-dimensional data sequentially to low-dimensional space. Finally, the low-dimensional features are used for visualization analysis and to train a classifier such as SVM (support vector machine) [11] and KNN (k-nearest neighbor) [12] and neural network classifiers [13].

With regard to the foregoing analysis, in conventional fault diagnosis methods, scholars have spent a large amount of time on feature extraction, feature selection, and dimensionality reduction, which are also complicated and long-standing tasks. In 2006, Hinton and Salakhutdinov published a paper in Science [14], which proposes two core points. First, an artificial neural network with multihidden layers possesses excellent feature-learning ability, and the acquired features provide a more intrinsic and abstract representation of the raw data. Second, layer-wise pretraining can effectively overcome the training difficulties of the deep neural network. Since then, research on deep learning in academia and industry has raised a large amount of attention. Researchers on speech recognition at Microsoft Research and Google decreased the speech recognition error rate by 20%–30% when they adopted deep neural networks (DNNs). In 2012, amazing results emerged in image recognition where the error rate was largely reduced from 26% to 15% in the ImageNet evaluation. In the same year, DNN was also applied to the prognosis of drug activity in pharmaceutical companies and achieved the world's best accuracy, which was featured in the New York Times.

Despite its success in speech, image, and video recognition, the application of deep learning in mechanical fault diagnosis has received very little research attention. Deep learning is quite different from traditional diagnosis methods that require complicated and time-consuming feature extraction work, which only needs simple data preprocessing. STFT (short-time Fourier transform) is a simple, easy-to-apply signal transformation method that can transform time-domain signals into time-frequency space. In this paper, a combination of deep learning networks and STFT is proposed to solve fault diagnosis problems. SAE (stacked sparse autoencoder), a neural network that consists of multiple layers of basic autoencoders in which the outputs of each layer are wired to the inputs of the successive layer, can learn higher order feature representations of input signals. In the deep-layer networks of SAE, raw data can be represented in a much better form, enabling the classifier to provide more accurate results even with fewer training examples or less labeled training data.

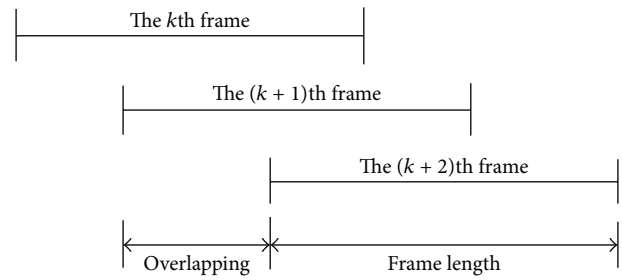


Figure 1: Data frames.

This paper is organized as follows. Section 2 introduces the basic principle of STFT. Section 3 proposes SAE based feature extraction. Section 4 describes softmax classifier-based pattern recognition. Section 5 outlines the implementation methodology of SAE with the softmax classifier and is followed by the conclusions in Section 6.

2. Time-Frequency Analysis of Sound Signals Using STFT

Fourier analysis decomposes a signal into its frequency components and determines their relative strengths. The Fourier transform is defined as

$$F(w) = \int_{-\infty}^{\infty} f(t)\, e^{-jw\tau} dt \longleftrightarrow f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(w)\, e^{jw\tau} dw. \tag{1}$$

This transform is applied to stationary signals whose properties do not evolve over time. When the signal is nonstationary, we can introduce a local frequency parameter in such a way that a local Fourier transform looks at the signal through a window over which the signal is approximately stationary. After multiplying it by a window, the signal is also truncated into short data frames, and to analyze the whole signal, the window is translated in time and then reapplied to the signal. The output of successive STFTs can provide a time-frequency representation of the signal [15].

Therefore, in this paper, a spectral analysis of sounds is performed by using STFT, in which the signal is divided into small sequential or overlapping data frames, as shown in Figure 1; then, FFT is applied to each data frame. The STFT positions a window function $\psi(t)$ at τ on the time axis and calculates the Fourier transform of the windowed signal as

$$F(w, \tau) = \int_{-\infty}^{\infty} f(t)\, \psi^*(t - \tau)\, e^{-jw\tau} dt. \tag{2}$$

The basic functions of this transform are generated by the modulation and transformation of the window function $\psi(t)$, where w and τ are the modulation and translation parameters, respectively [16]. Commonly used windows are the rectangular window, Hamming window, Hanning window, and Blackman window. The first two windows are described as follows in (3) and (4).

For a rectangular window of size N,

$$w(n) = \begin{cases} 1, & 0 \le n \le (N-1) \\ 0, & \text{others.} \end{cases} \tag{3}$$

For a Hamming window of size N,

$$w(n) = \begin{cases} 0.5\left(1 - \cos\dfrac{2\pi n}{N-1}\right), & 0 \le n \le (N-1) \\ 0, & \text{others.} \end{cases} \tag{4}$$

The rectangular window does not conform to the requirements for an excessively high side lobe, which leaks more energy. Therefore, the Hamming window is selected in this paper.

Given a signal $x(n)$, the discrete STFT for the frequency band k at time n is defined as

$$X_n\left(e^{j\omega_k}\right) = \sum_{m=-\infty}^{+\infty} x(m)\,\omega(n-m)\,e^{-j\omega_k m}, \tag{5}$$

where $\omega_k = 2\pi k/N$ is the frequency in radians; N is the number of frequency bands; $\omega(m)$ is the selected symmetric window of size L; and $L \le N$ if signal reconstruction is required.

It follows that (5) is equivalent to

$$X_n\left(e^{j\omega_k}\right) = e^{-j\omega_k n}\overline{X}_n(\omega_k), \tag{6}$$

where

$$\overline{X}_n(\omega_k) \sum_m x(n-m)\,\omega(m)\,e^{j\omega_k m} = x(n)\,h_k(n) \tag{7}$$

is the output of the kth complex band-pass filter, with impulse response $h_k(n)$ and center frequency f_k:

$$\begin{aligned} h_k(n) &= \omega(n)\,e^{j\omega_k n}, \\ f_k &= \frac{f_s k}{N}\ (\text{Hz}), \quad k = 0, 1, \ldots, N-1. \end{aligned} \tag{8}$$

According to $\omega_k = 2\pi k/N$ as above, plugging into (5) yields

$$\begin{aligned} X(n,k) &= X_n\left(e^{j\omega_k}\right) \\ &= \sum_{m=-\infty}^{+\infty} x(m)\,\omega(n-m)\,e^{-j2\pi km/N}. \end{aligned} \tag{9}$$

Here, $|X(n,k)|$ is the short-time spectral amplitude estimate of $x(n)$. The power spectrum density (PSD) function is defined as

$$P(n,k) = |X(n,k)|^2 = (x(n,k))\,x\left(\text{conj}\left(x(n,k)\right)\right). \tag{10}$$

$P(n,k)$ is a two-dimensional, nonnegative, and real-valued function. It is easily proven that $P(n,k)$ is only a Fourier transform (FT) of the short-time autocorrelation function of the signal. The spectrogram algorithm [17] is an analysis algorithm that produces a two-dimensional image representation of sounds. PSD is expressed as the Pseudo Color Map (PCM), in other words, a spectrogram with a time axis and frequency axis. This time-frequency spectrum, which is sometimes called visual language, shows the dynamic characteristics of the sounds and enjoys significant practical worth. The spectrogram is acquired as shown in Figure 2.

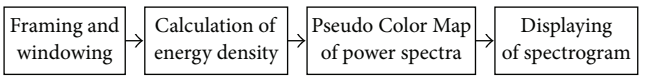


Figure 2: Generation of spectrogram.

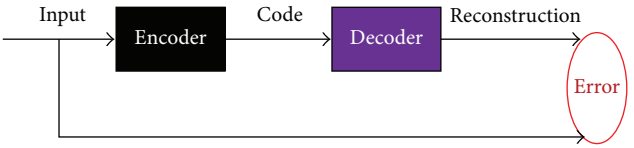


Figure 3: Schematic of an autoencoder.

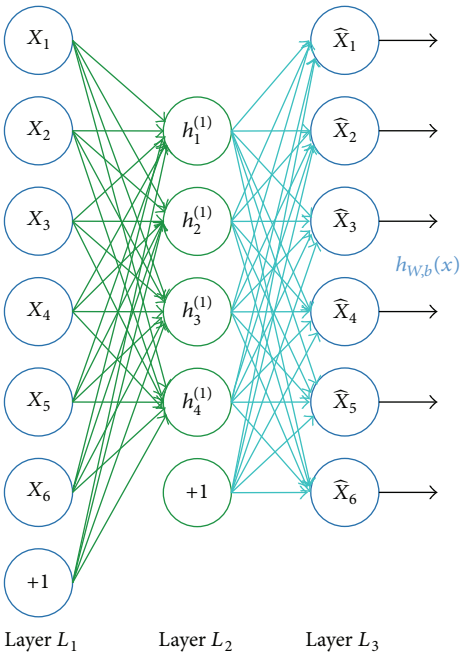


Figure 4: Structure of AE.

3. Feature Extraction Using SAE

3.1. Autoencoder. As depicted in Figure 4, an autoencoder that was first introduced by Rumelhart et al. is a special neural network with three layers. A trained autoencoder can compute the input's representation from which the original data can be reconstructed with as much accuracy as possible [18], as shown in Figure 3. Recently, autoencoders were used in deep architectures as an unsupervised learning algorithm [19, 20].

An autoencoder takes an input vector $x^{(i)} \in [0,1]$ that corresponds to the ith training example and first maps it to

the hidden layer $a \in [0,1]$ (a is the activation vector of the first hidden layer), through deterministic mapping:

$$a = f_\theta\left(x^i\right) = \text{sigmoid}\left(W \cdot x^i + b\right) \tag{11}$$

parameterized by $\theta = \{W, b\}$. The resulting latent representation a is then mapped back to a reconstructed vector $h_{W,b}(x^{(i)}) \in [0,1]$ in the input space [21, 22], as depicted in Figure 4:

$$h_{W,b}\left(x^{(i)}\right) = g_\theta(a) = \text{sigmoid}\left(W^T \cdot a + b^T\right). \tag{12}$$

(a) Cost Function of an Autoencoder. For a fixed training set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ of "m" training examples, the initial cost function is given by

$$\begin{aligned} J(W,b) &= \left[\frac{1}{m}\sum_{i=1}^{m} J\left(W,b;x^{(i)};y^{(i)}\right)\right] \\ &\quad + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2 \\ &= \left[\frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left\|h_{W,b}\left(x^{(i)}\right) - y^{(i)}\right\|^2\right] \\ &\quad + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2, \end{aligned} \tag{13}$$

where the first term in $J(W,b)$ is an average sum-of-squares error term. Here, W and b are the same as mentioned in (11) and (12). The second term is a regularization term or weight decay term, which tends to decrease the magnitude of the weights and helps prevent overfitting [22]. Here, $h_{W,b}(x^{(i)})$ is the hypothesis and λ is a weight decay parameter.

(b) Sparsity Constraint. The network architecture should be designed such that each training sample can be properly represented by a unique code and, therefore, can be reconstructed from the code with a small reconstruction error. This goal can be effectively achieved by making the code a discrete variable with a small number of different values or by making the code have a lower dimension than the input; alternatively, the code could be forced to be a "sparse" vector in which most of the components are zero [23].

Sparse overcomplete representations have a number of theoretical and practical advantages. Overcomplete representations have a basis vector that is greater than the dimensionality of the input. In particular, they have good robustness to noise [24]. We want hidden units to be inactive most of the time; that is, the outputs of the neurons should be close to zero for the sigmoid activation function. Then, we will write $a_j^{(2)}(x)$ to denote the activation of this hidden unit when the network is given a specific input x. Furthermore, $\widehat{\rho}_j = (1/m)\sum_{i=1}^{m}[a_j^{(2)}(x^{(i)})]$ denotes the average activation of hidden unit j (averaged over the training set). Then, the constraint $\widehat{\rho}_j = \rho$ is imposed, where ρ is a sparsity parameter, which is typically a small value close to zero; in our case, we used 0.1 [25].

To make the hidden unit's activation values penalty term will give reasonable results. The close to zero, an extra penalty term that penalizes $\widehat{\rho}_j$ deviating significantly from ρ is added in our optimization objective. Many choices of the penalty term will give reasonable results. The following is chosen [22]:

$$\sum_{j=1}^{s_2}\text{KL}\left(\rho \parallel \widehat{\rho}_j\right) = \sum_{j=1}^{s_2}\rho\log\frac{\rho}{\widehat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\widehat{\rho}_j}. \tag{14}$$

Here, s_2 is the number of neurons in the hidden layer, the index j sums up the hidden units in our network, and $\text{KL}(\rho \parallel \widehat{\rho}_j)$ is the Kullback-Leibler (KL) divergence between a Bernoulli random variable with a mean of ρ and a Bernoulli random variable with a mean of $\widehat{\rho}_j$.

On adding the penalty term, the overall cost function becomes

$$J(W,b)_{\text{sparse}} = J(W,b) + \beta\sum_{j=1}^{s_2}\text{KL}\left(\rho \parallel \widehat{\rho}_j\right). \tag{15}$$

The term β controls the weight of the sparsity penalty term [22].

3.2. Stacked Autoencoder. An efficient way to learn a complicated map is to combine a set of simpler models that are trained sequentially. The combined model performs a nonlinear transformation on the input vectors and produces an output that will be used as an input for the next model in the sequence. As shown in Figure 5, each autoencoder produces a more abstract representation of its input from the former autoencoder, and therefore, some of the stacked autoencoders can be pretrained to produce a high-level representation of the input data. In addition, fine-tuning the network parameters based on the pretraining can prevent its solution from getting stuck at a poor local minimum [22].

3.3. Unsupervised Feature Learning Using a Greedy Layer-Wise Approach. To learn the high-level features of the input in an unsupervised fashion, a greedy layer-wise approach is applied to train each autoencoder in turn. Formally, for a stacked autoencoder with n layers, $W^{(k,1)}$, $W^{(k,2)}$, $b^{(k,1)}$, and $b^{(k,2)}$ denote the parameters $W^{(1)}$, $W^{(2)}$, $b^{(1)}$, and $b^{(2)}$ for the kth autoencoder. Then, the encoding step for the stacked autoencoder is given by running the encoding step of each layer in forward order [22]:

$$a^{(l)} = f\left(z^{(l)}\right). \tag{16}$$

The decoding step is in reverse order:

$$\begin{aligned} a^{(n+l)} &= f\left(z^{(n+l)}\right), \\ z^{(n+l+1)} &= W^{(n-l,2)}a^{(n+l)} + b^{(n-l,2)}, \end{aligned} \tag{17}$$

where $a^{(n)}$ is an activation value of the deepest layer of the hidden units.

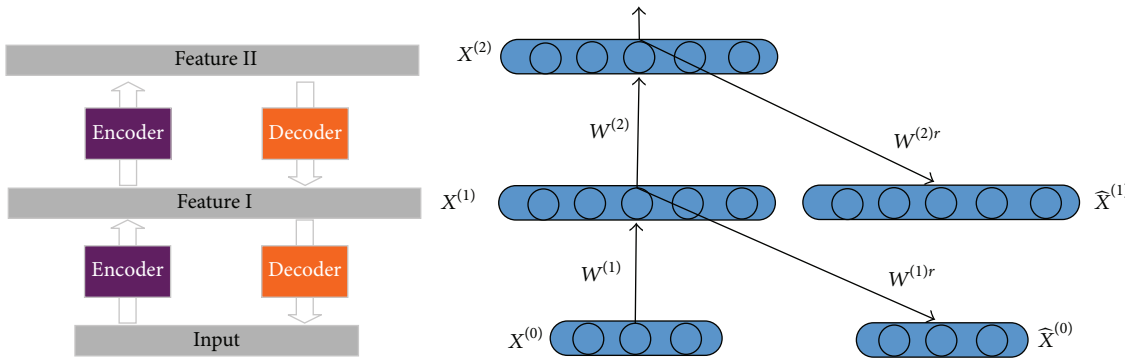


Figure 5: Schematic of SAE.

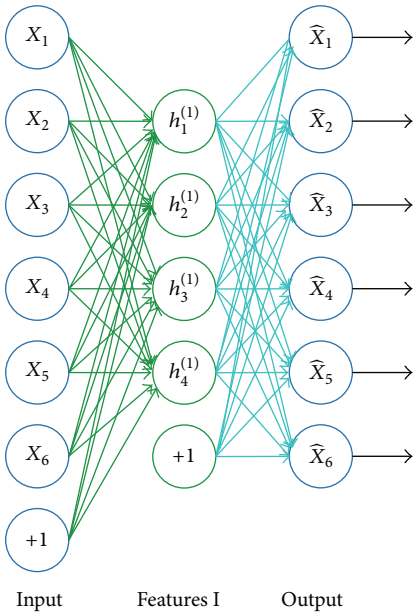


Figure 6: First-order representation.

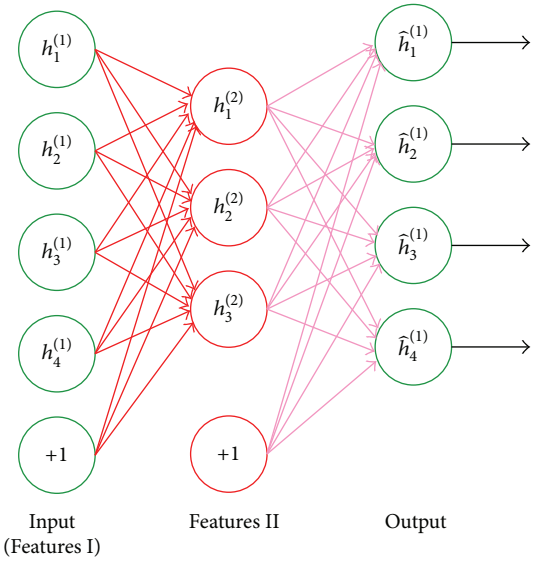


Figure 7: Second-order representation.

First, we train the first layer on raw input to obtain the parameters $W^{(1,1)}$, $W^{(1,2)}$, $b^{(1,1)}$, and $b^{(1,2)}$. We use the first layer to transform the raw input into a vector that consists of the activation of the hidden units A. We train the second layer on this vector to obtain the parameters $W^{(2,1)}$, $W^{(2,2)}$, $b^{(2,1)}$, and $b^{(2,2)}$. We repeat this sequence of actions for subsequent layers, using the output of each layer as input for the subsequent layer.

In this paper, a stacked autoencoder of two hidden layers is trained for the rolling bearing fault identification. First, a sparse autoencoder is trained to learn the first-order features $h^{(1)(k)}$ of the inputs $x^{(k)}$ (as shown in Figure 6).

Next, we feed the raw input into this trained sparse autoencoder, obtaining the primary feature activation $h^{(1)(k)}$ for each of the inputs $x^{(k)}$. We then use these primary features as "raw input" for another sparse autoencoder to learn the secondary features $h^{(2)(k)}$ on these primary features (as shown in Figure 7).

Next, we feed the primary features into the second sparse autoencoder to obtain the secondary feature activation $h^{(2)(k)}$ for each of the primary features $h^{(1)(k)}$ (which correspond to the primary features of the corresponding inputs $x^{(k)}$). The secondary features can then be treated as "raw input" to a softmax classifier, training it to map secondary features to the discrete digit labels (as shown in Figure 8).

Finally, two autoencoders and one classifier are wired together, building a stacked autoencoder with two hidden layers and a final softmax classifier layer that is capable of classifying the rolling bearing fault as desired (as shown in Figure 9).

3.4. Fine-Tuning Based on Back-Propagation. The greedy layer-wise approach pretrains the parameters of each layer individually while freezing the parameters for the remainder of the model. To produce better results, after this phase of training is completed, fine-tuning using backpropagation can be used to improve the results by tuning the parameters of all of the layers at the same time.

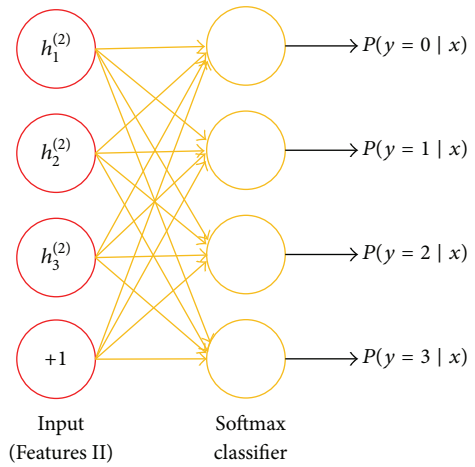


Figure 8: Softmax classifier.

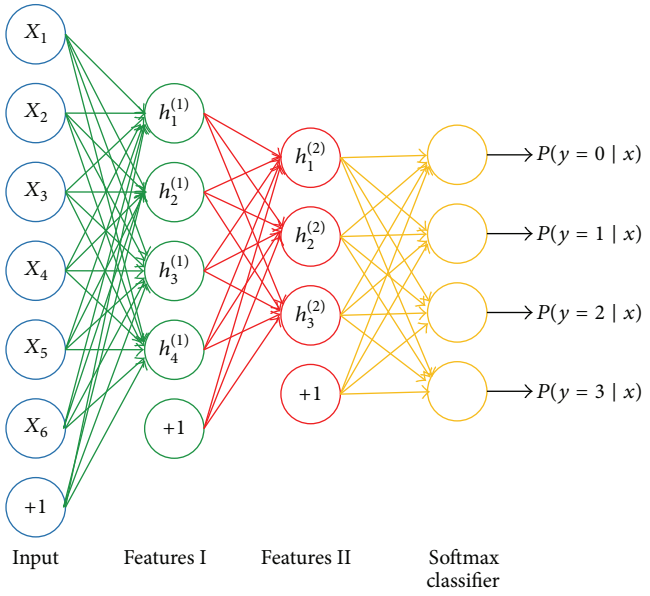


Figure 9: SAE with softmax classifier.

Fine-tuning of the weights of the network produces much better classification performance on the test data. It treats all of the layers of a stacked autoencoder as a single model, in such a way that in one iteration we can use the backpropagation algorithm to improve all of the weights in the stacked autoencoder. A summary of the fine-tuning with backpropagation using element-wise notation is given below [22]:

(1) Perform a feedforward pass, computing the activation values for layers L_1 and L_2, up to the output layer L_{nl}, using the equations that define the forward propagation steps.

(2) For the output layer (layer n_l), set

$$\delta^{(l)} = -\left(\nabla_{a^{n_l}} J\right) \cdot f'\left(z^{(n_l)}\right). \tag{18}$$

When using softmax regression, the softmax layer has $\nabla J = \theta^T (I - P)$, where I is the input labels and P is the vector of conditional probabilities.

(3) For $l = n_{l-1}, n_{l-2}, \ldots, 2$, we set

$$\delta^{(l)} = \left(\left(W^{(l)}\right)^T \delta^{(l+1)}\right) \cdot f'\left(z^{(l)}\right). \tag{19}$$

(4) Compute the desired partial derivatives:

$$\begin{aligned} \nabla_{W(l)} J\left(W, b; x, y\right) &= \delta^{(l+1)} \left(a^{(l)}\right)^T, \\ \nabla_{b^{(l)}} J\left(W, b; x, y\right) &= \delta^{(l+1)}. \end{aligned} \tag{20}$$

In this paper, we could consider the softmax classifier as an additional layer, but its derivation is calculated in a different way. Specifically, we consider the "last layer" of the network to be the features that are input into the softmax classifier. Therefore, the derivatives (in Step (2)) are computed using $\delta^{(l)} = -(\nabla_{a^{n_l}} J) \cdot f'(z^{(n_l)})$, where $\nabla J = \theta^T (I - P)$.

All of the weights and biases of the network in Figure 9 have been improved in the above four steps. The pretrained and fine-tuned SAE possesses the basic characteristics and performances of biological neural systems, in which different hidden layers extract different abstract characteristics, and the more abstract high-level feature has obvious superiority for classification. For a complex morphological and topological structure, SAE can provide powerful capacity in nonlinear modeling or prognostics and has several obvious advantages in large-scale parallelism, distributed processing, and self-organizing or self-learning.

4. Pattern Classification Based on Softmax Regression

A softmax classifier is a generalized logistic regression where the class labels can take on multiple values [22, 26].

For the training set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$, we have that $y^{(i)} \in \{1, 2, \ldots, k\}$. For a given test input x, the hypothesis estimates the probability $p(y = j \mid x)$ or each value of $j = 1, \ldots, k$, where k is the number of classes, that is, the estimate of the probability of the class label taking on each of the k different possible values. Thus, the hypothesis outputs a k-dimensional vector that gives the k estimated probabilities. Concretely, our hypothesis $h_\theta(x)$ takes the form [22]

$$\begin{aligned} h_\theta\left(x^{(i)}\right) &= \begin{bmatrix} p\left(y^{(i)} = 1 \mid x^{(i)}; \theta\right) \\ p\left(y^{(i)} = 2 \mid x^{(i)}; \theta\right) \\ \vdots \\ p\left(y^{(i)} = k \mid x^{(i)}; \theta\right) \end{bmatrix} \\ &= \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}, \end{aligned} \tag{21}$$

where $\theta_1, \theta_1, \ldots, \theta_k \in \Re^{n+1}$ are the model's parameters. Note that the term $\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}$ normalizes the distribution, in such a way that it sums to one. For convenience, we will also write θ to denote all of the parameters of our model. When softmax regression is implemented, it is usually convenient to represent θ as a k-by-$(n+1)$ matrix that is obtained by stacking up $\theta_1, \theta_1, \ldots, \theta_k$ in rows, and thus, $\theta = \begin{bmatrix} | & | & & | \\ \theta_1 & \theta_2 & \cdots & \theta_k \\ | & | & & | \end{bmatrix}^T$.

The cost function used by the softmax regression is given by

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\left\{y^i = j\right\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right] + \frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=0}^{n}\theta_{ij}^2. \tag{22}$$

In the equation above, $1\{\cdot\}$ is the indicator function, which means that $1\{\text{a true statement}\} = 1$ and $1\{\text{a false statement}\} = 0$. With this weight decay term (for any $\lambda > 0$), the cost function $J(\theta)$ is strictly convex and is guaranteed to have a unique solution. The Hessian is invertible, and because $J(\theta)$ is convex, algorithms such as gradient descent and L-BFGS (limited-memory Broyden-Fletcher-Goldfarb-Shanno) are guaranteed to converge to the global minimum [22].

One can show that the derivative of $J(\theta)$ is

$$\nabla_{\theta j} J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[x^{(i)}\left(1\left\{y^{(i)} = j\right\} - p\left(y^{(i)} = j \mid x^{(i)}; \theta\right)\right)\right] + \lambda\theta_j. \tag{23}$$

By minimizing $J(\theta)$ with respect to θ, we will have a working implementation of softmax regression.

5. Rolling Bearing Fault Diagnosis Based on STFT and SAE

5.1. The Proposed Fault Diagnosis Scheme. In this section, a novel rolling bearing fault diagnosis method based on STFT and SAE is proposed, and Figure 10 briefly depicts the overall scheme for the fault identification.

(1) Recording and Preprocessing. Sound signals are acquired by a recording device, and each sample is approximately one minute in duration. Furthermore, the outliers in the data are removed or replaced.

(2) The STFT Analysis of the Sound Signals. In this step, the spectrogram algorithm is used to obtain the spectra and spectrum matrixes of the sounds, whose related parameter settings will be detailed in Section 5.2.4.

(3) Data Normalization and Selection. For convenient subsequent data processing, spectrum matrixes are normalized by column into gray-value matrixes. Min-max normalization, called deviation normalization, is conducted in this paper, which maps each element of the matrixes to an integer value from 0 to 255. The transform function can be written as follows:

$$x^* = \frac{x - \min}{\max - \min} \times 255. \tag{24}$$

Here, min is the minimum, while max is the maximum in a column.

After the modulus of each spectrogram element is first determined, normalization is performed. Certain data from each column in the center of the matrixes is finally chosen to be inputs of the SAE network.

(4) Fault Feature Extraction Based on SAE. The SAE of two hidden layers can be trained by spectrogram data in an unsupervised way, which is a deep learning process. An eventual representation of the raw data is achieved by layer-by-layer learning, where the outputs of the first hidden layer become the inputs of the second hidden layer.

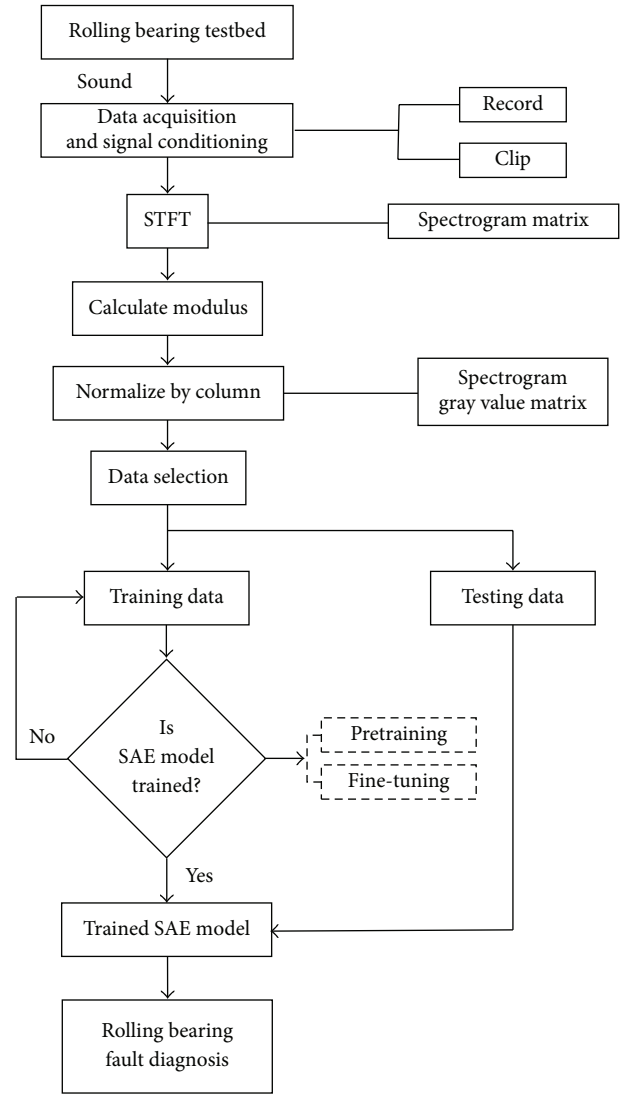


Figure 10: Flowchart of rolling bearing fault diagnosis.

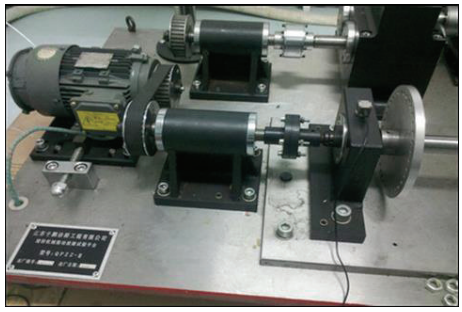


Figure 11: Rolling bearing test stand.

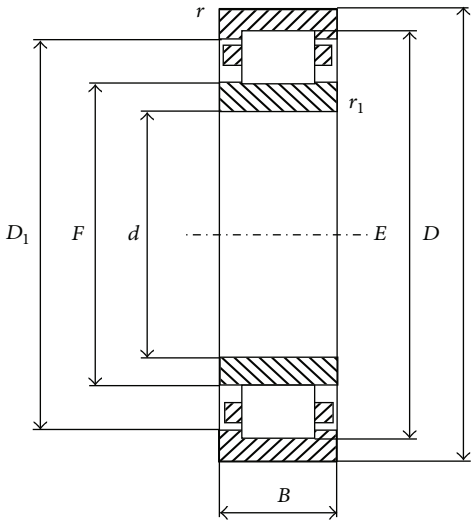


Figure 12: Profile of the rolling bearing.

(5) Fault Modes Classification Based on a Softmax Classifier. First, the eventual fault feature representation from the SAE is transformed into inputs of the softmax classifier. Through minimizing the cost function, the probability of each classification result will be calculated. Consequently, if one fault probability emerges as the maxima, then the input data can be identified as that fault.

5.2. Experimental Data Analysis

5.2.1. Data Acquisition. As shown in Figure 11, the test stand consists of a motor, a belt transmission, a coupling, and two bearing housings. The test bearings support a shaft with a turntable. In the test, four N205 bearings with different faults are installed and tested in turn, among which there are one normal bearing and three fault bearings of one inner-race fault, one outer-race fault, and one rolling parts fault. The structure and basic structural parameters of the tested rolling bearings are depicted, respectively, in Figure 12 and in Table 1. The sound data were acquired using a recorder, which was attached on a steel scaffold near the bearing block but without contacting the test stand, at 44,100 samples per second under the rotational speed of 1200 rpm.

5.2.2. The STFT Analysis. Spectrogram in the Matlab 8.1 function library is employed to extract the time-frequency information in the sounds, and the spectrograms of the four fault modes are described in Figure 13.

Table 1: Rolling bearing parameters.

d/mm	D/mm	D_0/mm	d_0/mm	α/deg	z/piece
25	52	38.5	7.5	0	12

The parameters in the table are as follows: d: inner diameter; D: outer diameter; D_0: pitch diameter; d_0: rolling parts diameter; α: contact angle; and z: number of rolling parts.

Table 2: Settings of STFT.

Window function	Window size	N_{overlap}	N_{fft}	f_s
Hamming window	44100	44000	44100	44100

The parameters in the table are as follows: N_{overlap}: the number of overlapping points; N_{fft}: the number of fast Fourier transform points; and f_s: the sampling frequency.

Table 3: Settings of SAE.

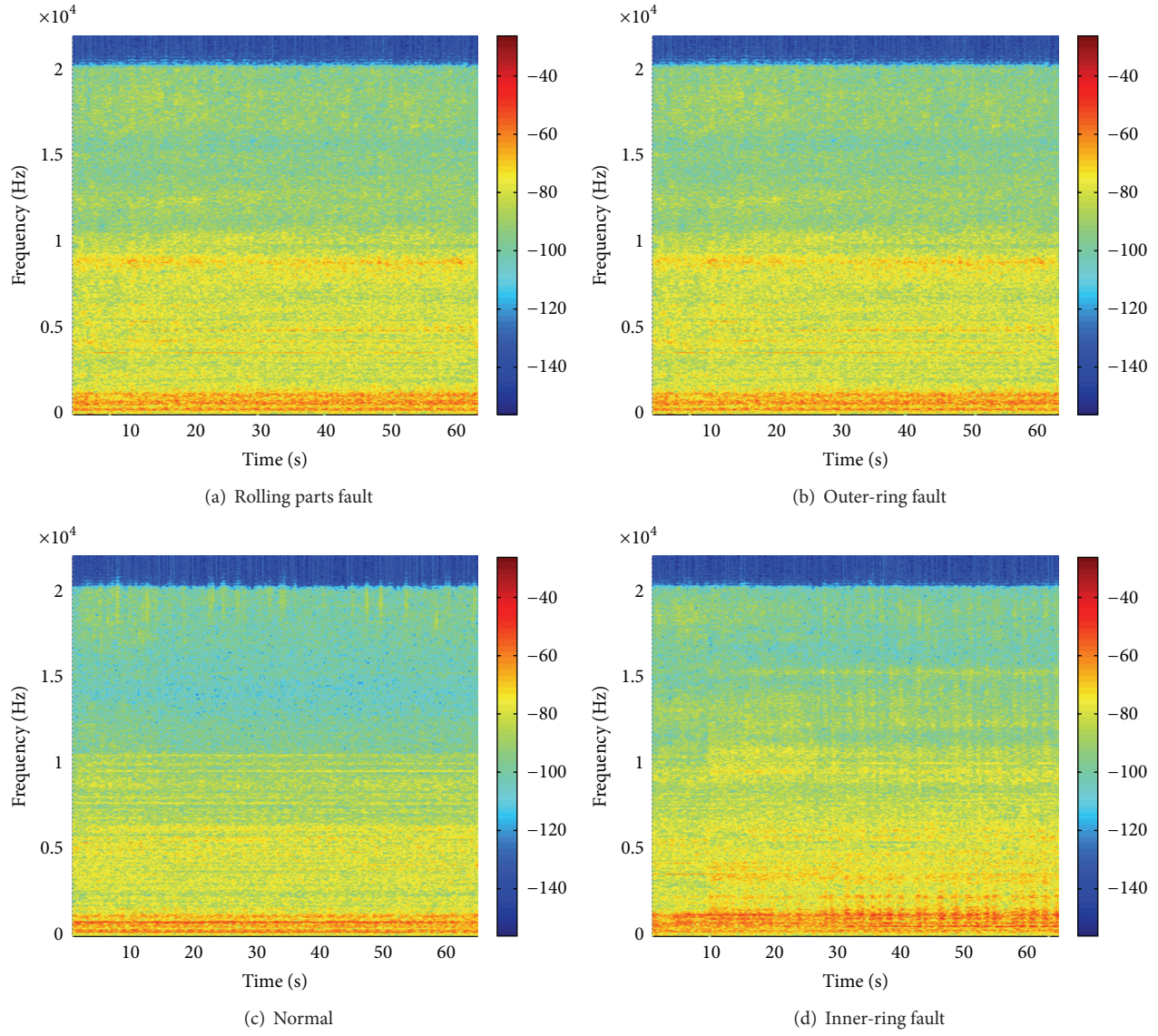
Layer 1	The number of input layer nodes	6500
	The number of hidden layer nodes	1000
	Sparsity	0.1
	Sparsity penalty factor of the loss function	3
	Weight decay factor of the loss function	0.003
Layer 2	The number of input layer nodes	1000
	The number of hidden layer nodes	100
	Sparsity	0.1
	Sparsity penalty factor of the loss function	3
	Weight decay factor of the loss function	0.003
Layer 3	The number of input layer nodes	100
	The number of hidden layer nodes	10
	Sparsity	0.1
	Sparsity penalty factor of the loss function	3
	Weight decay factor of the loss function	0.003
Softmax classifier	The number of input layer nodes	10
	The number of output layer nodes	4

5.2.3. Data Normalization and Selection. Min-max normalization is performed in this section to map each element of the spectrogram matrixes to an integer from 0 to 255. The acquired gray images are shown in Figure 14, and a method for selecting the SAE network's inputs is proposed in Figure 15.

5.2.4. The Experiment on Fault Modes Classification. According to the proposed diagnosis scheme, an SAE with a softmax classifier network is proposed to automatically identify the faults after simple data preprocessing. The experimental parameters are set up as follows.

(1) Settings of STFT. The parameter settings of the STFT are shown in Table 2.

(2) Settings of SAE. The parameter settings for SAE are detailed in Table 3.



(a) Rolling parts fault

(b) Outer-ring fault

(c) Normal

(d) Inner-ring fault

Figure 13: Spectrograms of the four fault modes.

Table 4: Data information.

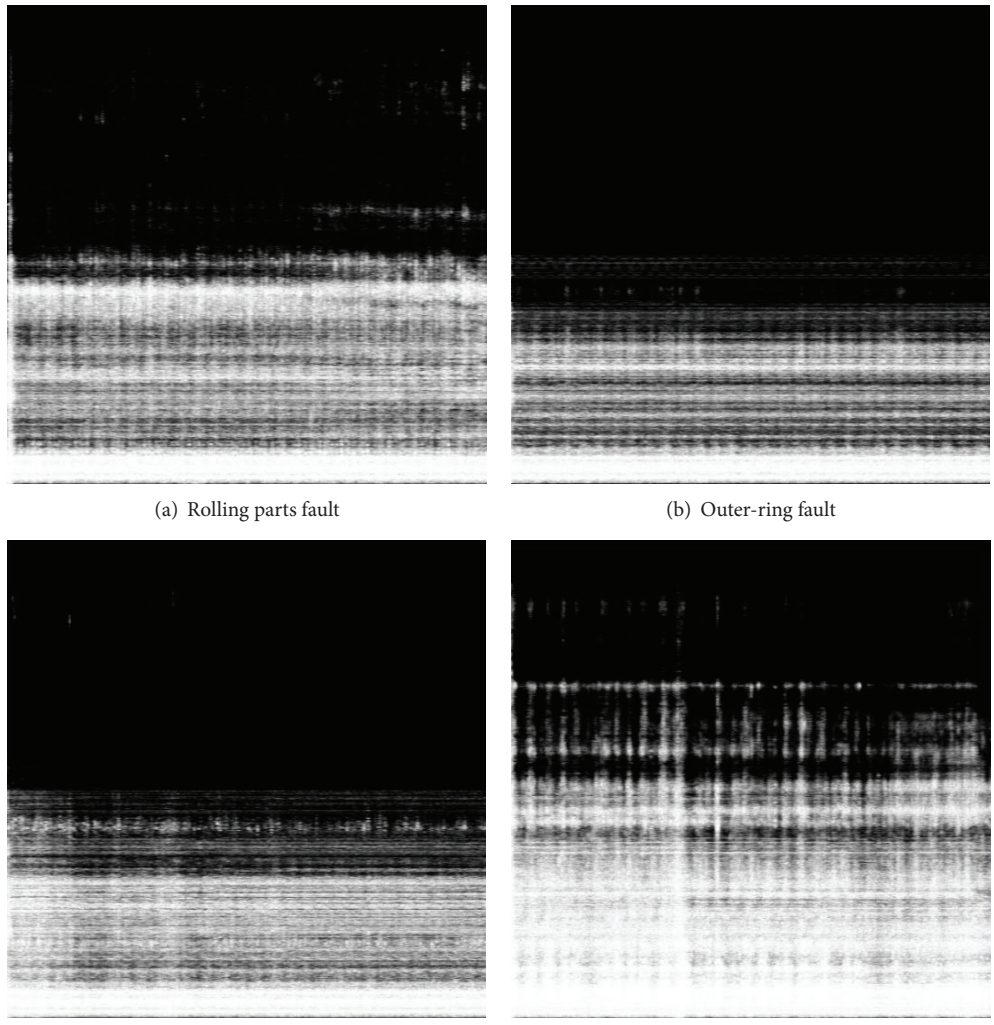
Group index	Normal	Inner-race fault	Outer-race fault	Rolling parts fault	Total
G1	3000	3000	3000	3000	12000
G2	3000	3000	3000	3000	12000

5.2.5. Results of the Test Analysis. Under the above settings, SAE with a softmax classifier is trained and then used to recognize the faults of the rolling bearings by the sound signals. The proposed approach can be verified by a two-set cross-validation method, where the data are divided in half; one-half is selected to be the training data, and the other half is selected to be the testing data. An introduction to the data is shown in Table 4.

First, the proposed method is applied to identify whether a testing bearing is a failure or not. The experimental results are given in Table 5. From the chart, the classification accuracy of each validation is higher than 97% after the networks are fine-tuned, and the average could reach up to

Table 5: Identification results on two fault modes.

Cross-validation	Classification accuracy after fine-tuning	Average
G1 for training and G2 for testing	98.35%	97.84%
G2 for training and G1 for testing	97.33%	



(a) Rolling parts fault (b) Outer-ring fault

(c) Normal (d) Inner-ring fault

Figure 14: Gray images of four fault modes.

Table 6: Identification results on four fault modes.

Cross-validation	Normal	Inner-race fault	Outer-race fault	Rolling parts fault	Average
G1 for training and G2 for testing	100%	93.4%	100%	100%	95.68%
G2 for training and G1 for testing	100%	99.1%	90.24%	100%	

97.84%, which demonstrates that the method has excellent and powerful capability for use in health detection.

Next, this method is used to recognize the faults from the normal, inner-race fault, outer-race fault, and rolling bearing parts fault bearings. The diagnosis results are shown in Table 6. From the table, we can determine that the method has good recognition performance on four fault modes, and it increases the average recognition rate to 95.68%.

5.3. Comparisons of the Proposed Method with EMD-TEO and SAE Using Vibration Signals. In this subsection, based on vibration signals, EMD-TEO and SAE are also employed to diagnose rolling bearing faults. The analysis results are illustrated in detail.

5.3.1. EMD-TEO Based on Vibration Signals. A fault diagnosis method based on empirical mode decomposition (EMD), Teager Energy Operator (TEO), and the softmax classifier is described as follows:. First, vibration signals are decomposed into several Intrinsic Mode Components (IMFs) by using EMD. Second, TEO is used to extract the instantaneous amplitudes of the IMFs. Third, several amplitude ratios in the frequency spectra of demodulated IMFs are extracted as fault feature vectors, and then, Principal Components Analysis

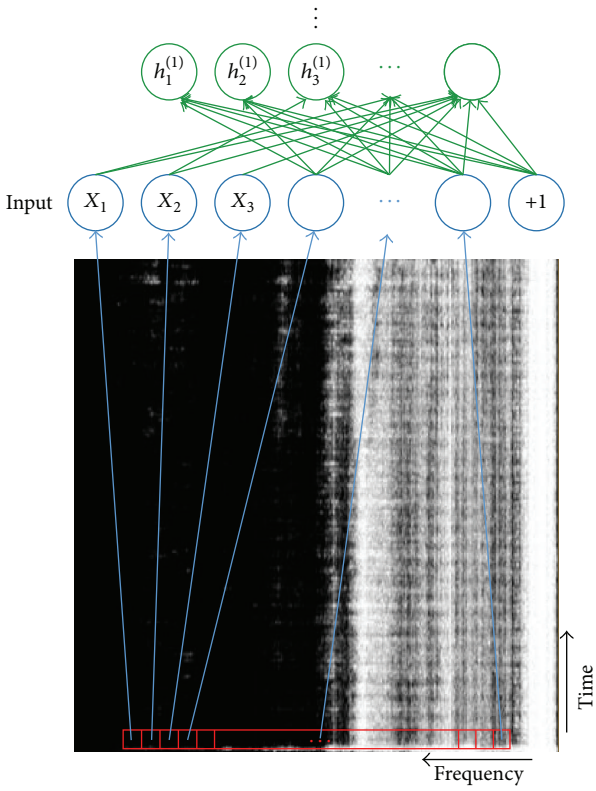


Figure 15: Data selection method.

Table 7: Fault identification results using EMD-TEO.

	Normal	Inner-race fault	Outer-race fault	Ball fault	Average
Accuracy	100%	90%	90%	100%	95%

(PCA) is applied for dimensionality reduction. Finally, these feature vectors are taken as inputs to train and test the softmax classifier. The diagnosis results are shown in Table 7.

5.3.2. SAE Based on Vibration Signals. In this part, SAE with a softmax classifier network is utilized to automatically identify faults based on vibration signals.

(1) Settings of SAE. The parameter settings of SAE are shown in Table 8.

(2) Identification Results. Under the above settings, SAE with a softmax classifier is trained and then used to recognize the faults of rolling bearings by vibration signals. The identification results are shown in Table 9.

5.3.3. Comparison Conclusion. From Tables 6, 7, and 9, SAE combined with STFT using sound signals can realize equal fault identification performance with traditional EMD-TEO and SAE based on vibration signals, but the EMD-TEO method spends too much time on artificially extracting the fault features, and specific instruments are required to acquire the vibration signals.

Table 8: Settings of SAE.

Layer 1	The number of input neurons	128
	The number of hidden neurons	64
	Sparsity	0.1
	Sparsity penalty factor of the loss function	3
	Weight decay factor of the loss function	0.003
Layer 2	The number of input neurons	64
	The number of hidden layer neurons	32
	Sparsity	0.1
	Sparsity penalty factor of the loss function	3
	Weight decay factor of the loss function	0.003
Layer 3	The number of input neurons	32
	The number of hidden neurons	16
	Sparsity	0.1
	Sparsity penalty factor of the loss function	3
	Weight decay factor of the loss function	0.003
Softmax classifier	The number of input neurons	16
	The number of output neurons	4

Table 9: Fault identification results using SAE.

	Normal	Inner-race fault	Outer-race fault	Ball fault	Average
Accuracy	100%	95.53%	91.17%	98.46%	96.29%

6. Conclusions

Because traditional feature extraction methods are time-consuming and require more experience, a novel rolling bearing fault diagnosis method based on STFT and a deep learning network is proposed. By STFT, the original sound signals are mapped into time-frequency space first. Then, SAE is proposed to automatically extract the intrinsic fault features of the rolling bearings. Last, softmax regression is utilized to recognize the fault modes of the feature vectors. Comparison results reveal that the proposed method outperforms traditional fault diagnosis method using vibration signals and realize equal fault identification performance with SAE based on vibration signals.

The proposed method is much easier to apply widely in a highly automated industry because it is data-driven without human interference. In particular, for large and nonstandard bearings, this method can be implemented to analyze fault locations and, thus, help operators and manufacturers to replace the faulty part. Due to the favorable robustness and diagnostic performance, this method can also be easily applied for fault diagnosis in a wide spectrum of machines.

Limited by the consumption of computer resources, to some extent, the proposed method might not be sufficiently satisfactory in "real time." As expected, STFT and spectrogram functions quickly consume a vast amount of memory for their extensive matrix operations. Furthermore, the accuracy and efficiency of the proposed method would probably be influenced by changes in the working conditions, such as a changed rotation speed. Therefore, further study can be conducted on decreasing the consumption of computer

memory and increasing its adaptability to new working conditions in advance.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant nos. 51605014, 61074083, 51575021, and 51105019) as well as the Technology Foundation Program of National Defense (Grant no. Z132013B002).

References

[1] K. Shibata, A. Takahashi, and T. Shirai, "Fault diagnosis of rotating machinery through visualisation of sound signals," *Mechanical Systems and Signal Processing*, vol. 14, no. 2, pp. 229–241, 2000.

[2] J. Lin, "Feature extraction of machine sound using wavelet and its application in fault diagnosis," *NDT & E International*, vol. 34, no. 1, pp. 25–30, 2001.

[3] D. Yu, J. Cheng, and Y. Yang, "Application of EMD method and Hilbert spectrum to the fault diagnosis of roller bearings," *Mechanical Systems and Signal Processing*, vol. 19, no. 2, pp. 259–270, 2005.

[4] W. Y. Liu, W. H. Zhang, J. G. Han, and G. F. Wang, "A new wind turbine fault diagnosis method based on the local mean decomposition," *Renewable Energy*, vol. 48, pp. 411–415, 2012.

[5] H. Liu, X. Wang, and C. Lu, "Rolling bearing fault diagnosis based on LCD–TEO and multifractal detrended fluctuation analysis," *Mechanical Systems and Signal Processing*, vol. 60, pp. 273–288, 2015.

[6] W. Sun, J. Chen, and J. Li, "Decision tree and PCA-based fault diagnosis of rotating machinery," *Mechanical Systems and Signal Processing*, vol. 21, no. 3, pp. 1300–1317, 2007.

[7] S. W. Choi, C. Lee, J.-M. Lee, J. H. Park, and I.-B. Lee, "Fault detection and identification of nonlinear processes based on kernel PCA," *Chemometrics and Intelligent Laboratory Systems*, vol. 75, no. 1, pp. 55–67, 2005.

[8] Z. Li, X. Yan, C. Yuan, J. Zhao, and Z. Peng, "The fault diagnosis approach for gears using multidimensional features and intelligent classifier," *Noise & Vibration Worldwide*, vol. 41, no. 10, pp. 76–86, 2010.

[9] Z. Wei, Z. Weijia, and L. Bin, "Fault diagnosis approach based on fractal dimension LLE and Fisher discriminant," *Chinese Journal of Scientific Instrument*, vol. 31, no. 2, pp. 325–333, 2010.

[10] G. B. Wang, X. Q. Zhao, and Y. H. He, "Fault diagnosis method based on supervised incremental local tangent space alignment and SVM," *Applied Mechanics and Materials*, vol. 34-35, pp. 1233–1237, 2010.

[11] L. Shuang and L. Meng, "Bearing fault diagnosis based on PCA and SVM," in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '07)*, pp. 3503–3507, IEEE, Harbin, China, August 2007.

[12] B. Bagheri, H. Ahmadi, and R. Labbafi, "Application of data mining and feature extraction on intelligent fault diagnosis by artificial neural network and k-nearest neighbor," in *Proceedings of the 19th International Conference on Electrical Machines (ICEM '10)*, pp. 1–7, IEEE, Rome, Italy, September 2010.

[13] C. Chen and C. Mo, "A method for intelligent fault diagnosis of rotating machinery," *Digital Signal Processing*, vol. 14, no. 3, pp. 203–217, 2004.

[14] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[15] M. K. Kıymık, I. Güler, A. Dizibüyük, and M. Akin, "Comparison of STFT and wavelet transform methods in determining epileptic seizure activity in EEG signals for real-time application," *Computers in Biology and Medicine*, vol. 35, no. 7, pp. 603–616, 2005.

[16] F. Jurado and J. R. Saenz, "Comparison between discrete STFT and wavelets for the analysis of power quality events," *Electric Power Systems Research*, vol. 62, no. 3, pp. 183–190, 2002.

[17] L. Deng and I. Kheirallah, "Dynamic formant tracking of noisy speech using temporal analysis on outputs from a nonlinear cochlear model," *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 5, pp. 456–467, 1993.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[19] Y. Bengio and Y. LeCun, "Scaling learning algorithms towards AI," *Large-Scale Kernel Machines*, vol. 34, no. 5, pp. 1–41, 2007.

[20] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *The Journal of Machine Learning Research*, vol. 10, pp. 1–40, 2009.

[21] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103, ACM, Helsinki, Finland, July 2008.

[22] Unsupervised Feature Learning and Deep Learning (UFLDL), 2011, http://deeplearning.stanford.edu/wiki/index.php/UFLDL_Tutorial.

[23] Y. L. Boureau, S. Chopra, and Y. Lecun, "A unified energy-based framework for unsupervised learning," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 371–379, 2007.

[24] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 315–323, Ft. Lauderdale, Fla, USA, April 2011.

[25] N. K. Verma, V. K. Gupta, M. Sharma, and R. K. Sevakula, "Intelligent condition based monitoring of rotating machines using sparse auto-encoders," in *Proceedings of the IEEE Conference on Prognostics and Health Management (PHM '13)*, pp. 1–7, June 2013.

[26] K. Duan, S. S. Keerthi, W. Chu, S. K. Shevade, and A. N. Poo, "Multi-category classification by soft-max combination of binary classifiers," in *Multiple Classifier Systems*, vol. 2709 of *Lecture Notes in Computer Science*, pp. 125–134, Springer, Berlin, Germany, 2003.