Journal of Computational Information Systems 11: 24 (2015) 7723–7730
Available at http://www.Jofcis.com

Fault Diagnosis in High-speed Train Running Gears with Improved Deep Belief Networks ⋆

Jipeng XIE [1], Yan YANG [1,∗], Hao WANG [1], Tianrui LI [1], Weidong JIN [2]

[1] *School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China*

[2] *School of Electrical Engineering, Southwest Jiaotong University, Chengdu 610031, China*

Abstract

This paper explores the Deep Belief Networks (DBNs) in the application of high-speed train vibration signals processing. Firstly, a method based on DBNs is proposed. The vibration signals are preprocessed by Fast Fourier Transform (FFT) and then the FFT coefficient-vectors set the states of the visible units of DBNs. DBNs can be trained greedily, layer by layer to learn high-level features, using a model referred to Restricted Boltzmann Machine (RBM). For more efficient learning, a new model called K-DBNs is proposed by combining the advantages of K-Nearest Neighbor (KNN) and DBNs. Finally, another new method optimizing each layer in deep network is introduced by integrating unsupervised learning with supervised learning. Experiments on real datasets and simulation datasets confirm that the proposed methods may learn useful high-level features and diagnose different faults of high-speed train.

Keywords: Deep Belief Networks; Feature Extraction; Fault Diagnosis; K-Nearest Neighbor

1 Introduction

With the development of high-speed train, its security issues attract more and more attention. Fault diagnosis is a significant part for the security issues of high-speed train. And, how to extract features quickly and efficiently from these massive vibration data and diagnose faults accurately is a hot topic. In [1], Lei et al. presented a method which is applied on fault feature extraction of EEMD for fault diagnosis of rotating machinery and locomotive roller bearings. Yao et al. studied fault diagnosis of train bearings based on harmonic wavelet envelope [2]. Dash et al. analyzed the GA-fuzzy controller for fault diagnosis in Cracked Structure [3]. Zhao et al. introduced a method based on EMD and Fuzzy Entropy to extract the feature, and BP neural network was used for the fault diagnosis of high-speed train [4]. In [5], feature selection was done with wavelet entropy, and then SVM was used for fault diagnosis. Nevertheless, a lot of problems appear in

⋆Project supported by the National Nature Science Foundation of China (Nos. 61134002, 61170111, 61175047 and 61572407) and the Research Fund of Traction Power State Key Laboratory of Southwest Jiaotong University (No. 2012TPL_T15).

∗Corresponding author.
Email address: yyang@swjtu.edu.cn (Yan YANG).

DOI: 10.12733/jcis15C0233
December 15, 2015

practical applications. For example, it is difficult to know whether the fault features (e.g. fault feature extraction of EMD or EEMD) are stable; the signal processing methods (e.g. harmonic wavelet envelope) are restricted by the number of samples.

It is noteworthy that deep learning [6] is one of the most powerful representation learning techniques and DBNs [7] are pioneers in building deep architectures. Recently, DBNs have already been applied to hand-written character recognition, speech recognition, health diagnosis of complex engineered systems and natural language understanding. To the best of our knowledge, research-related about DBNs applied to fault diagnosis of high-speed train is not much.

In our previous works [8], we used DBNs on FFT to automatically extract deep features from vibration signals under different conditions and then recognized the faults. This paper is an extended version of our works. Its contributions include two ways. On the one hand, considering the inconformity of data types and dimensions caused by the difference type of sensor and other factors, one model called K-DBNs is proposed by combining the advantages of KNN and DBNs. On the other hand, RBM is trained in our previous works with an unsupervised learning method so that it may loss much label information. Besides, it is likely to cause the problem of overfitting that just simply taking supervised learning method. Thus, a model called Sup-DBNs is proposed through integrating unsupervised learning with supervised learning in training RBM [9].

The rest of this paper is organized as follows: In Section 2, it describes the model of DBNs including K-DBNs and Sup-DBNs on FFT for fault diagnosis. Section 3 gives experimental setup and demonstrates the results of each experiment. Finally, Section 4 concludes this paper.

It is worth noting that we have explained massive necessary theoretical knowledge and related works in our pervious works [8]. Simple but not lost necessity, we also introduce some preparatory theories for this article in following sections.

2 Training Deep Belief Networks

2.1 Learning features by DBNs from vibration signals

In order to apply DBNs to learn features from high-speed train vibration signals, the FFT coefficient-vectors which were normalized to [0, 1] in dimension are used to set the states of the visible units of the lower layer of DBNs. And then, the DBNs are pre-trained with the training sets in an unsupervised learning way among "*vis-top*" layers for learning high-level features. The pre-training does not use any information about the class labels. After a stack of RBMs have been trained, n "*softmaxed*" label units ($[e^{x_i}/\sum_i e^{x_i}]$, where x_i is the weighted input produced by the feature activations plus a bias term) are connected to the "*top*" layer and the whole network is fine-tuned using conjugate gradient descent in cross-entropy error ($[-\sum_i p_i \log \hat{p}_i - \sum_i (1-p_i)\log(1-\hat{p}_i)]$, where p_i is the value of FFT coefficient point i and $\hat{p}_i$ is the value of its reconstruction). The model is shown in Fig. 1.

2.2 Training the model of K-DBNs

In DBNs, it just used unsupervised learning method to learn higher level features, and then performed global fine-tuning by using back-propagation or up-down algorithm. While, K-DBNs considers about K nearest instances and then performs fault diagnosis after learning feature

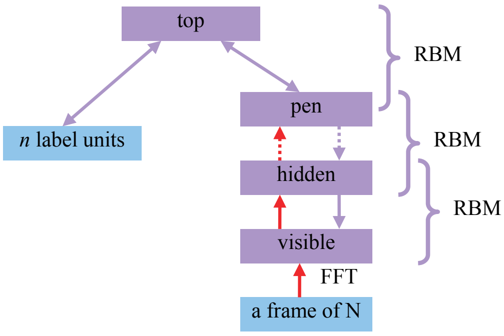

Fig. 1: DBNs model on vibration data and labels

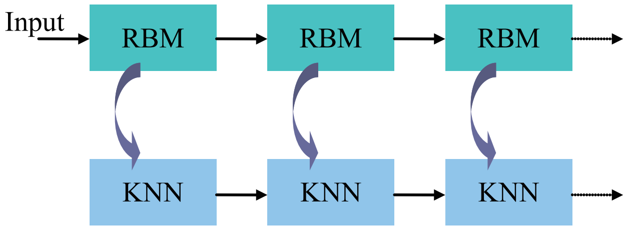

Fig. 2: K-DBNs model combing KNN and DBNs

representation by DBNs in each hidden layer. Fig. 2 gives the model of K-DBNs. It is divided into two processes. Assuming that there are **n** instances, it trains as same as DBNs using **m** instances in process one. During another process, it gets K_1 nearest neighbor instances for a new instance of the remaining **t(n-m)** instances in the first hidden layer, counts the frequency of each class label and transmit these nearest instances to next hidden layer. Then, it gets K_2 nearest neighbor instances in the second hidden layer, and processes as the last step processing. Finally, K-DBNs trains via layer by layer and gets the only class of the new instance. Algorithm 1 gives a detailed description of the algorithm.

Algorithm 1 K-Nearest Neighbor Deep Belief Networks

Input: Sample data **X** having n data objects and the class numbers **C**
Output: Accuracy **Acc** of fault diagnosis

(1) Extract multilayer features by **r** RBM models and determine the weight $\{\boldsymbol{w_1}, \boldsymbol{w_2}, ...\}$ of layer to layer

a. Randomly select **m** sample in **X**, train the first RBM to learn the feature $\boldsymbol{F_1}$ and weight $\boldsymbol{w_1}$, and transmit $\boldsymbol{F_1}$ to next RBM.

b. Use $\boldsymbol{F_1}$ to train the second RBM, learn the feature $\boldsymbol{F_2}$ and $\boldsymbol{w_2}$.

c. Iteratively, learn the **r** layer feature $\boldsymbol{F_r}$ and weight $\boldsymbol{w_r}$.

d. Finally, feedback to fine-tune the whole network weight.

(2) Fix the weight $\{\boldsymbol{w_1}, \boldsymbol{w_2}, ...\}$, use the multilayer feature from process 1 to class the remaining sample by KNN

a. Select the remaining **t(n-m)** sample, learn feature $\boldsymbol{f_1}$ by weight $\boldsymbol{w_1}$ in $\boldsymbol{F_1}$, get $\boldsymbol{K_1}$ nearest sample for $\boldsymbol{f_1}$, count the frequency of these $\boldsymbol{K_1}$ nearest sample and assign the class label for instance based on KNN.

b. If these class labels are consensus, return to step a.; else, transmit $\boldsymbol{f_1}$ and sample $\boldsymbol{K_1}$ to next hidden layer.

c. Learn the feature $\boldsymbol{f_2}$ by weight $\boldsymbol{w_2}$, get $\boldsymbol{K_2}$ nearest sample for $\boldsymbol{f_2}$ in $\boldsymbol{K_1}$ nearest sample, then assign the class label as a. step processing.

d. If these class labels are consensus, return to step a.; else, transmit $\boldsymbol{f_2}$ and sample $\boldsymbol{K_2}$ to next hidden layer.

e. Iteratively, perform layer by layer. Finally, when **t** samples all have their class label, align them with the original class label and count **Acc**.

2.3 Training the model of Sup-DBNs

In machine learning, some algorithm like RBM can effectively learn the abstract and the high-level features hiding in original data by unsupervised learning method. However, they may loss the existing label information in training process. On the other side, it is likely to appear over-fitting that simply taking supervised learning method like Linear Discriminant Analysis (LDA). Integrating unsupervised learning with supervised learning seems to be advisable. Before doing this work, we introduce feature selection firstly, Fisher Ratio (F-Ratio) [11], which may effectively remove the irrelevant and redundant features and choose the most effective features from the original features.

Assuming that a total of K classes, the class j having N_j feature vectors, then F-Ratio is:

$$F_i = \frac{B_i}{W_i}, \tag{1}$$

where B_i is inter-class variance and W_i is intra-class variance in feature i, which are given by:

$$B_i = \frac{1}{K}\sum_{j=1}^{K}(\mu_{ij} - \mu_i)^2, \; W_i = \frac{1}{K}\sum_{j=1}^{K} W_{ij}; \tag{2}$$

where μ_i is grand mean of feature i, μ_{ij} and W_{ij} are mean and variance of feature i in class j.

It evaluates the feature of the hidden layer by F-Ratio and removes redundant features. Then, it minimizes the class error rate in each feature layer by Logistic Regression (LR). The model of Sup-DBNs is shown in Fig. 3. This model is also composed of multilayer and trained via layer by layer. For illustrating the model more clearly, we analyze one single layer which is shown in 4. Assuming that X is the original sample data and Y is sample label which set as the input of visible layer and R is the refactoring data learning from hidden layer by DBNs. In order to use these label information, we add F-Ratio and LR to RBM for minimizing the cost error between X and R by iterative learning. The cost error is given by:

$$E_c = \sum_{j=1}^{l} L \cdot \log \sum_{i=1}^{m} \frac{x_i}{\sum_{i=1}^{m} x_i}, \;\; as\; x_i = \exp\left(F_{x\times l} \cdot Q_{f\times l} + c_l\right). \tag{3}$$

where Q_l is the connected weight between feature layer and label layer. l is the numbers of sample class and c_l is bias in label layer.

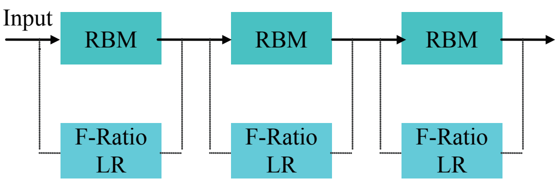

Fig. 3: Sup-DBNs model. Integrating Unsup-learning with Sup-learning by F-Ratio and LR

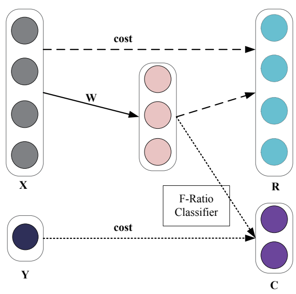

Fig. 4: One single layer of Sup-DBNs. W is the weight between v and h, C is classing result

3 Experimental Setup & Results

3.1 Experimental setup

1) Real datasets

The tested shock absorbers consist of air spring, transverse shock absorber and anti-yaw shock absorber. Real datasets under four conditions were acquired by the sensors at different locations of train, such as the corbel (car-body forepart). The four conditions are the normal train, without anti-yaw shock absorber, air spring failure, and without transverse shock absorber respectively, which are described in our previous work in [8]. The sampling frequency is 234 Hz and the speed of train is 80, 120, 140, 160, 200 km/h.

A real datasets (RDS) contains four subsets corresponding to four different conditions. And the RDS80, RDS120, RDS140, RDS160 and RDS200 correspond to 80 km/h, 120 km/h, 140 km/h, 160 km/h and 200 km/h, respectively. Each real data subset, which is a vibration signal containing 14,600 sampling points, corresponds to one of the four conditions and a certain speed. We divided the 14,600 sampling points into short frames of 280 points, 52 cases. Thus, a RDS had 208 cases in total: 52 cases being condition-1, 52 cases for condition-2, 52 cases for condition-3 and the rest 52 cases belonging to condition-4.

2) Simulation datasets

A simulation datasets (SDS) containing seven subsets was generated from simulation system (SIMPACK and Track Spectrum). Mock locations: the corbel (car-body forepart); sampling frequency: 243Hz; mock conditions: seven conditions are the normal train, without anti-yaw shock absorber, air spring failure, without transverse shock absorber and the rest three conditions are compound fault which correspond to condition-2, 3, condition-2, 4 and condition-3, 4, respectively. We divided each simulation data subset into short frames of 280 at set intervals. A SDS had 14,000 cases in total: 2,000 cases being condition-1, 2,000 cases for condition-2, 2,000 cases for condition-3, 2,000 cases for condition-4, 2,000 cases for condition-5, 2,000 cases for condition-6 and the rest 2,000 cases belonging to condition-7.

3.2 Experimental results

All experiments were performed with RDS for a total of 208 cases and SDS for a total of 14,000 cases except specially stated. The experiments are as follows.

1) Experiment 1: time domain analysis and frequency domain analysis

Time-frequency analysis, which is commonly used in analyzing the feature of running gears, provides joint information in time and frequency domain. In this way, processing non-stationary and time-varying vibration signals will be achievable.

In analysis of time domain of the original signal (the speed of train is 120 km/h and the sampling time is 10 seconds), as shown in Fig. 5(a), The distribution of the waveform is complex. While, in frequency domain, the low frequencies from 0 to 30 Hz are observed, and the waveforms in frequency domain mainly distribute in a low-frequency range which includes frequencies up to 6 Hz, as shown in Fig. 5(b). Thus, the amplitudes of the low frequencies from 0 to 5.8733 Hz (the number of frequency points is 100) are used to set the states of the visible units of all DBNs in the following experiments.

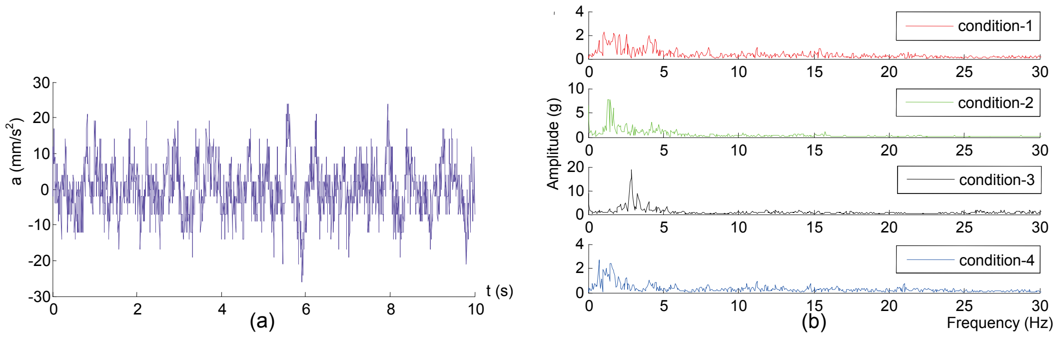

Fig. 5: The original signal: (a) time domain, (b) frequency spectrum

2) Experiment 2: fault diagnosis performance of the DBNs on FFT

The number of bits which is taken to specify a data-vector determines how much constraint each training case imposes on the parameters of the DBNs [10]. We want to explore the performance of the DBNs in the fault diagnosis by varying the number of layers and the size of hidden units. For simplicity, we used the same size for every hidden layer in networks. All DBNs were pre-trained with a fixed method using stochastic gradient decent with a mini-batch size of 1,260 training cases. For each RBM, we used 50 epochs to train it with a fixed learning rate of 0.1. According to Fig. 1, seven output units were connected to the "*top*" layer and the whole networks were fine-tuned. After several epochs of fine-tuning, the accuracy rate on the training data reached a pre-specified threshold value and the fine-tuning was stopped. The test accuracy rates at that point are shown in Fig. 6(a) and Fig. 6(b). Fig. 6(a) shows the fault diagnosis accuracy rates for different number of layers. The noticeable trend in Fig. 6(a) is that a DBN with four layers shows a good performance. Fig. 6(b) shows the fault diagnosis accuracy rates for different size of hidden units. Overall, as shown in the Fig. 6(b), the recognition rate remains fairly constant with the size of hidden units increasing in each layer.

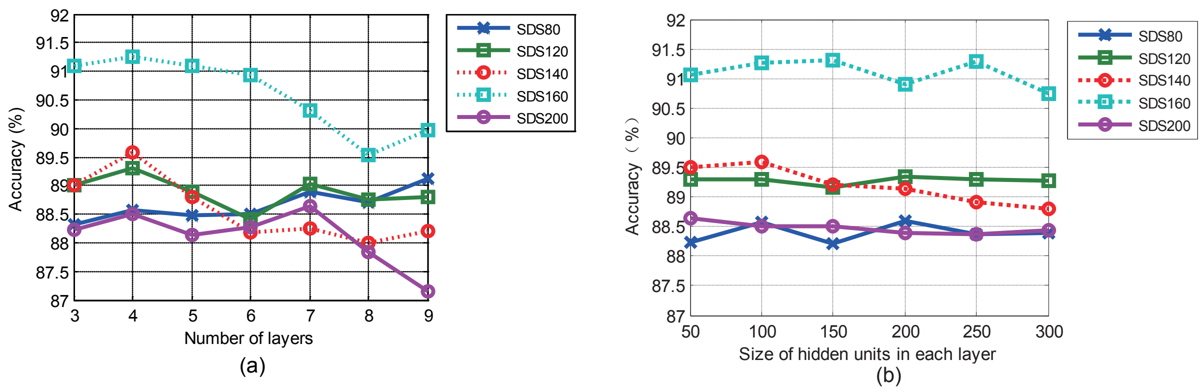

Fig. 6: Fault diagnosis accuracy rate on the SDS on the basis of (a)the number of layers, using 100 hidden units per hidden layer, (b)the size of hidden units in each layer, totally two hidden layers

3) Experiment 3: fault diagnosis performance of the K-DBNs on FFT

In order to explore the performance of K-DBNs, we collected real datasets at the train corbel floor, three-axis gearbox and axle box. In contrast, these datasets were respectively trained by DBNs and K-DBNs. The experiment results are shown in Fig. 7.

As shown in Fig. 7(a), the performance of two algorithms for failure recognition is similar when they test the first dataset. Fig. 7(b) and Fig. 7(c) show that the performance of K-DBNs is significantly more superior than DBNs when they testing the second and third dataset (note that different channel data may produce slight difference in experiment performance). The experiment results have no significance superior on some channel data (e.g. the first dataset). And in this case, it is hard to say that the improved models are significantly better than the original DBN model. However, it also gets remarkable achievement in some aspect. So to speak, the K-DBNs is no weaker than the original DBN moble as a whole.

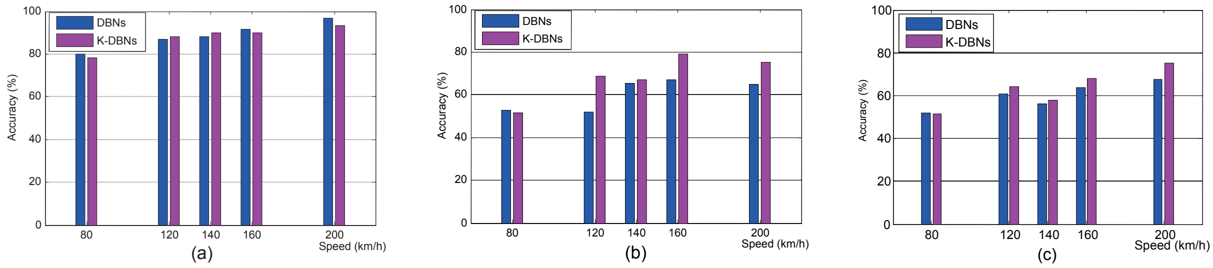

Fig. 7: Fault accuracy rate on RDS on (a)train corbel floor, (b)three-axis gearbox, (c)axle box

4) Experiment 4: fault diagnosis performance of the Sup-DBNs on FFT

The SDS (SDS120, SDS140, SDS160 and SDS200) were used as test set to explore the performance of improved algorithm. We respectively pre-trained (280)-100-100-50-20 nets using Unsup-DBNs (our previous method) and Sup-DBNs to learn features in high-layer. Then, these learning features were set as the input of SVM which processed fault diagnosis. In the training process, we set the iteration times of each layer as 1 to 70. Fig. 8 gives the accuracy rate of fault diagnosis.

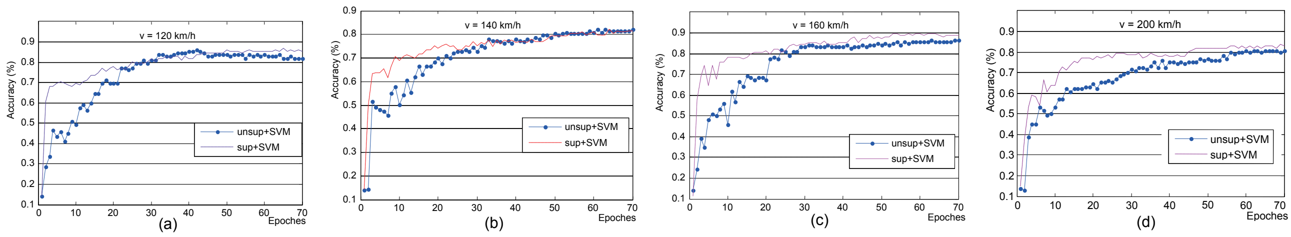

Fig. 8: Fault accuracy rate on SDS ((a)SDS120, (b)SDS140, (c)SDS160, (d)SDS200)

According to these results, Unsup-DBNs and Sup-DBNs all reach to a stable state after 40 times training. It is worth noted that the accuracy rate of Sup-DBNs on SDS160 and SDS200 is higher than Unsup-DBNs and the convergence speed is faster too. Especially, the remarkable performance are acquired by Sup-DBNs before convergence. Comparing the experiment results, algorithm-improved achieves a better effect on fault diagnosis as a whole.

4 Conclusion

In this paper, we explore DBNs in circumstances of high-speed train vibration. Vibration signals were analyzed in time and frequency domain in experiment 1 and the frequency of the signals range mainly concentrates on the low frequency band. Furthermore, the parameters of DBNs, the number of layers and the size of hidden units were explored in experiment 2. DBNs with four layers showed a good performance on SDS and the recognition rate remained fairly constant with the size of hidden units increasing in each layer. In addition, the results of experiment 3 and 4 confirm that the improved methods called K-DBNs and Sup-DBNs provide better quality compared to our previous learning method. These results also motivate further research with deep neural networks applied to high-speed train signal analysis. In our future work, we will focus on the investigation of multi-channel data fusion based on DBNs for fault diagnosis in high-speed train running gears.

References

[1] Y. Lei, Z. He, Y. Zi, Application of the EEMD method to rotor fault diagnosis of rotating machinery, Mechanical Systems and Signal Processing, 23 (2009), pp. 1327-1338.

[2] D. Yao, L. Jia, M. Li, et al., Harmonic Wavelet Envelope Method Applied in Railway Bearing Fault Diagnosis, Journal of Engineering Science and Technology Review, 6(2), (2013), pp. 24-28.

[3] A. K. Dash, D. R. Parhi, Analysis of an Intelligent Hybrid System for Fault Diagnosis in Cracked Structure, Arabian Journal for Science and Engineering, 39(2), (2014), pp. 1337-1357.

[4] J. J. Zhao, Y. Yang, et al., Application of Empirical Mode Decomposition and Fuzzy Entropy to High-Speed Rail Fault Diagnosis, Foundations of Intelligent Systems, Springer Berlin Heidelberg, 2014, pp. 93-103.

[5] N. Qin, W. D. Jin, J. Huang, et al., High Speed Train Bogie Fault Signal Analysis Based on Wavelet Entropy Feature, Advanced Materials Research, 753 (2013), pp. 2286-2289.

[6] X. L. Zhang, Learning deep representation without parameter inference for nonlinear dimensionality reduction, arXiv preprint arXiv: 1308. 4922, 2013.

[7] G. E. Hinton, S. Osindero, Y. W. Teh, A fast learning algorithm for deep belief nets, Neural computation, 18(7), (2006), pp. 1527-1554.

[8] J. P. Xie, Y. Yang, et al., Learning Features form High Speed Train Vibration Signals with Deep Belief Networks, International Joint Conference on Neural Networks, 2014, pp. 2205-2210.

[9] J. B. Zhang, G. J. Tian, et al., Supervised deep learning with auxiliary networks, Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 353-361.

[10] G. Hinton, A practical guide to training restricted Boltzmann machines, Neural Networks: Tricks of the Trade, Springer Berlin Heidelberg, 2012, pp. 599-619.

[11] C. C. Huang, W. Gong, et al., Research of speaker recognition based on the weighted fisher ratio of MFCC, International Conference on Mechatronic Sciences, Electric Engineering and Computer, 2013, pp. 904-907.