2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Learning Features from High Speed Train Vibration Signals with Deep Belief Networks

Jipeng Xie
School of Information Science and Technology
Southwest Jiaotong University
Chengdu, China
jipengxie@163.com

Yan Yang*
School of Information Science and Technology
Southwest Jiaotong University
Chengdu, China
yyang@swjtu.edu.cn

Tianrui Li
School of Information Science and Technology
Southwest Jiaotong University
Chengdu, China
trli@swjtu.edu.cn

Weidong Jin
School of Electrical Engineering
Southwest Jiaotong University
Chengdu, China
wdjin@swjtu.edu.cn

***Abstract*—Feature extraction is one of key steps in fault diagnosis for High Speed Train (HST). In this work, we present a method that can automatically extract high-level features from HST vibration signals and recognize the faults. The method is composed of a Deep Belief Network (DBN) on Fast Fourier Transform (FFT) of vibration signals. DBNs can be trained greedily, layer by layer, using a model referred to as a Restricted Boltzmann Machine (RBM). The real data sets and simulation data sets of HST vibration signals are selected in experiments. First, the vibration signals are preprocessed by FFT. Then, the FFT coefficient-vectors are used to set the states of the visible units of DBNs. Finally, *n* label units are connected to the "*top*" layer of the DBNs to identify different faults. The experimental results show that the method may learn useful high-level features from vibration signals and diagnose the different faults of HST.**

Keywords*—*feature extraction; Deep Belief Network; Fast Fourier Transform; vibration signals

I. Introduction

With the development of High Speed Train (HST), its security issues have received much attention recently. Vibration data collected by sensors reflects the operation condition and it is concerned with the safety of HST. How to analyze and take advantage of this data adequately and how to extract features with respect to the operation condition of HST is a hot topic. In [1, 2], Lei et al. presented a method which is applied on fault feature extraction of Ensemble Empirical Mode Decomposition (EEMD) for fault diagnosis of rotating machinery and locomotive roller bearings. Chen et al. proposed an approach of EEMD-1.5 dimension spectrum for fault feature extraction [3]. Yao et al. studied fault diagnosis of train bearings based on harmonic wavelet envelope [4]. Dash et al. analyzed the Genetic Algorithm and fuzzy (GA-fuzzy) controller for fault diagnosis in Cracked Structure [5]. Raveendran et al. presented a dynamic model-based approach to quantify the severity of upstream mechanical equipment faults [6]. However, a lot of problems appear in practical applications. For example, it is difficult to know whether the fault features (e.g. fault feature extraction of EEMD) are stable; the signal processing methods (e.g. harmonic wavelet envelope) are restricted by the number of samples. To address these problems, we built a model on the simulation data sets of HST vibration signals.

There have been many works in automatic feature extraction and fault recognition so far. Zhao et al. introduced a method based on Empirical Mode Decomposition (EMD) and Fuzzy Entropy to extract the features of HST vibration signals, and Back Propagation (BP) neural network was used as the model for the fault diagnosis of HST [7]. Li proposed a Self-Organized Map (SOM) neural network together with Principal Component Analysis (PCA) for the fault diagnosis of rolling bearing [8]. In [9], feature selection was done with wavelet entropy, and then Support Vector Machine (SVM) was used as the model of fault recognition.

Deep learning [10] is one of the most powerful representation learning techniques and Deep Belief Networks(DBNs) [11, 12] are pioneers in building deep architectures. DBNs can be considered as a highly complex nonlinear feature extractor in which each hidden layer learns to represent features that acquire higher order correlations in the original input data. DBNs have already been applied to hand-written character recognition [13], speech recognition [14], 3-D object recognition [15], extracting road maps from cluttered aerial images [16] and information retrieval [17].

In our approach, we use the DBNs on FFT to extract deep

*Corresponding author: yyang@swjtu.edu.cn. This work was supported by the National Natural Science Foundation of China (Nos. 61134002, 61170111 and 61175047), the Research Fund of Traction Power State Key Laboratory of Southwest Jiaotong University (No. 2012TPL_T15) and the Fundamental Research Funds for the Central Universities (No. SWJTU11ZT08).

features automatically from vibration signals of HST under different conditions. The vibration signals, which were captured from different locations of train under the different health conditions, are first preprocessed by FFT. Then, DBNs are pre-trained in an unsupervised manner for learning high-level features. Finally, the label layer is symmetrically connected to the "*top*" layer of DBNs and the whole networks are fine-tuned in a supervised way. Moreover, 10-fold cross-validation [18] is used to evaluate the recognition performance of our method.

The rest of this paper is organized as follows: In Section II, we introduce the DBNs model and describe the method of using a DBN on FFT for learning features from HST vibration signals. In Section III, experiments evaluation is provided including the experimental setup and the results of the experiments. Finally, Section IV concludes the paper.

II. TRAINING DEEP BELIEF NETWORKS

A. Restricted Boltzmann Machine

The simple learning model used in the undirected view is called a Restricted Boltzmann Machine (RBM) [19]. It is a bipartite graph in which visible units **v**, representing observations, are connected via symmetrically undirected weights to hidden units **h**. It is restricted in the situation that there are no visible-visible or hidden-hidden connections. In the simplest type of RBM (See Fig. 1), both the hidden and visible units are binary and stochastic. RBM has an efficient training procedure which makes it suitable as building blocks in training DBNs.

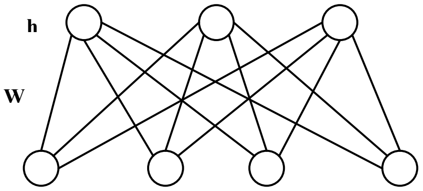


Fig. 1. RBM model. The bottom layer represents a vector of binary visible units **v** and the top layer represents a vector of binary hidden units **h**. **W** is the weight between **v** and **h**.

For a Bernoulli-Bernoulli RBM [25], a joint configuration, ($\mathbf{v}\in\{0,1\}^n$, $\mathbf{h}\in\{0,1\}^m$) of the visible and hidden units has an energy given by:

$$E(\mathbf{v},\mathbf{h};\theta) = -\mathbf{v}^{\mathrm{T}}\mathbf{W}\mathbf{h} - \mathbf{b}^{\mathrm{T}}\mathbf{v} - \mathbf{a}^{\mathrm{T}}\mathbf{h} = -\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij} v_i h_j - \sum_{i=1}^{n} b_i v_i - \sum_{j=1}^{m} a_j h_j, \tag{1}$$

where $\theta=\{\mathbf{W},\mathbf{b},\mathbf{a}\}$ are the model parameters: W_{ij} represents the symmetric interaction weight between visible unit i and hidden unit j ; b_i , a_j are their biases; v_i , h_j are the binary states of visible unit i and hidden unit j . The joint distribution over the visible and hidden units is defined by:

$$P(\mathbf{v},\mathbf{h};\theta) = \frac{1}{\mathrm{Z}(\theta)}\exp(-E(\mathbf{v},\mathbf{h};\theta)), \tag{2}$$

$$Z(\theta) = \sum_{\mathbf{v}}\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h};\theta)), \tag{3}$$

where $Z(\theta)$ is the "partition function". The probability that the model assigns to a visible vector **v** is:

$$P(\mathbf{v};\theta) = \frac{1}{\mathrm{Z}(\theta)}\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h};\theta)). \tag{4}$$

Owing to the special bipartite structure of RBM, the hidden units can be explicitly marginalized out:

$$P(\mathbf{v};\theta) = \frac{1}{\mathrm{Z}(\theta)}\sum_{\mathbf{h}}\exp(\mathbf{v}^{\mathrm{T}}\mathbf{W}\mathbf{h} + \mathbf{b}^{\mathrm{T}}\mathbf{v} + \mathbf{a}^{\mathrm{T}}\mathbf{h}) \tag{5}$$

The conditional distributions over hidden units **h** and visible vector **v** can be derived from (2) and are given by logistic functions:

$$P(\mathbf{v}\mid\mathbf{h};\theta) = \prod_{i}^{n} p(v_i\mid\mathbf{h}), \tag{6}$$

$$P(\mathbf{h}\mid\mathbf{v};\theta) = \prod_{j=1}^{m} p(h_j\mid\mathbf{v}), \tag{7}$$

$$p(v_i=1\mid\mathbf{h}) = \delta(b_i + \sum_{j=1}^{m} W_{ij}h_j), \tag{8}$$

$$p(h_j=1\mid\mathbf{v}) = \delta(a_j + \sum_{i=1}^{n} W_{ij}v_i), \tag{9}$$

where $\delta(x)=1/(1+\exp(-x))$ is the logistic function. The derivative of the log-likelihood with respect to the model parameters θ can be obtained from (4):

$$\frac{\partial\log P(\mathbf{v};\theta)}{\partial W} = E_{P_{data}}\langle\mathbf{v}\mathbf{h}^{T}\rangle - E_{P_{model}}\langle\mathbf{v}\mathbf{h}^{T}\rangle, \tag{10}$$

$$\frac{\partial\log P(\mathbf{v};\theta)}{\partial a} = E_{P_{data}}\langle\mathbf{h}\rangle - E_{P_{model}}\langle\mathbf{h}\rangle, \tag{11}$$

$$\frac{\partial\log P(\mathbf{v};\theta)}{\partial b} = E_{P_{data}}\langle\mathbf{v}\rangle - E_{P_{model}}\langle\mathbf{v}\rangle, \tag{12}$$

where $E_{P_{data}}\langle\cdot\rangle$ denotes an expectation with respect to the data distribution $P_{data}(\mathbf{h},\mathbf{v};\theta) = P(\mathbf{h}\mid\mathbf{v};\theta)P_{data}(\mathbf{v})$, with $P_{data}(\mathbf{v}) = \frac{1}{N}\sum_k \delta(\mathbf{v}-\mathbf{v}_k)$ representing the empirical distribution, and $E_{P_{model}}\langle\cdot\rangle$ is an expectation with respect to the distribution defined by the model, as in (2). In fact, learning is done by a very simple learning rule for performing an approximation to the gradient of a different objective function, called the "Contrastive Divergence" (CD) [20]:

$$\Delta W_{ij} = \alpha(E_{P_{data}}\langle\mathbf{v}\mathbf{h}^{T}\rangle - E_{P_T}\langle\mathbf{v}\mathbf{h}^{T}\rangle), \tag{13}$$

where α is the learning rate and P_T represents a distribution defined by running a Markov chain that uses alternating Gibbs sampling, initialized by setting the binary states of the visible units to be the same as a data-vector, for T full steps. Setting $T=q$ (or CD$_q$) will be used to denote learning using q full steps of alternating Gibbs sampling. For training RBM, the CD learning with $T=1$ (or CD$_1$) has been shown to work quite well [21]. But of course, RBM learns better if more steps (e.g. $T=10$) of alternating Gibbs sampling are used [22].

For Gaussian–Bernoulli RBM [25], the energy of a joint configuration is

$$E(\mathbf{v},\mathbf{h};\theta) = -\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}\frac{v_i}{\sigma_i}h_j - \sum_{i=1}^{n}\frac{(v_i-b_i)^2}{2\sigma_i^2} - \sum_{j=1}^{m} a_j h_j, \tag{14}$$

where σ_i is the standard deviation of the Gaussian noise for visible unit i.

Since there are no visible-visible connections, the conditional distribution $p(\mathbf{v}|\mathbf{h},\theta)$ is factorial and is given by

$$p(\mathbf{v}|\mathbf{h},\theta) = N(b_i + \sum_{j}^{m} W_{ij}h_j, \sigma_i^2 = 1), \tag{15}$$

where $N(\mu,\sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2. Apart from these differences, the inference and learning rules for a Gaussian-Bernoulli RBM are the same as for a Bernoulli-Bernoulli RBM, though the learning rate α needs to be smaller. Bernoulli-Bernoulli RBMs are used to build DBNs in the paper.

B. Deep Belief Networks

DBNs are probabilistic generative models, containing many layers of hidden variables. Firstly, a RBM is used to learn the weights for one layer of a neural network in an unsupervised pattern. Once the weights are learned, the outputs from this layer are used as input features to train another RBM that learns a higher level feature representation. This greedy, layer-wise training process is both fast and effective [12]. After a stack of RBMs has been trained, the layers are connected together to form what is referred to as a DBN and a sets of learned initial weights are produced. Supervised fine-tuning is then performed using the DBN and the initial weights via back-propagation or up-down algorithm to retrain the weights [23-26].

C. Learning Features by DBNs from Vibration Signals

In order to apply DBNs to learn features from HST 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 set 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. 2. In Fig. 2, "*a frame of N*" denotes a data-vector including N data points; "*n*" denotes the number of faults; "*pen*" is the hidden layer which is closest to the "*top*" layer (note that the "*top*" layer is also a hidden layer).

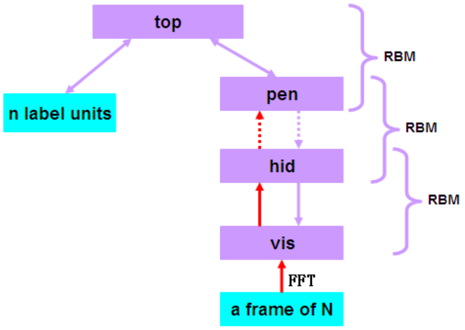


Fig. 2. FFT-DBNs model

III. EXPERIMENTAL EVALUATIONS

A. Experimental Setup

1) Real Data Sets

The vibration mechanical model of locomotive shock absorbers is shown in Fig. 3. The tested shock absorbers consist of air spring, transverse shock absorber and anti-yaw shock absorber, as shown in Fig. 3. Real data sets under four conditions were acquired by the sensors at different locations of train, such as the corbel (car-body forepart). The four conditions are described in Table I. and the parameters in the experiments are listed in Table II.

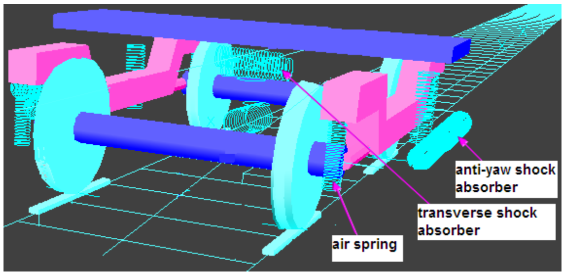


Fig. 3. The vibration mechanical model

TABLE I
A DESCRIPTION OF THE FOUR CONDITIONS

Type	Condition
condition-1	the normal train
condition-2	without anti-yaw shock absorber
condition-3	air spring failure
condition-4	without transverse shock absorber

TABLE II
A DESCRIPTION OF PARAMETERS

Parameter	Value
Sensor type	LC0709-2
Sampling frequency	243 Hz
Train's speeds	80, 120, 140, 160, 200 km/h

A Real Data Set (RDS) contains four subsets corresponding to four different conditions. 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 (e.g., one real data subset including 14,600 sampling points was acquired by the sensor fixed on the floor of train under condition-1 and the train was running at a speed of 80 kilometers per hour). We divide the

14,600 sampling points into short frames of 280, so a subset of 52 cases is produced. Thus, a RDS including four subsets have 208 cases: 52 cases being condition-1, 52 cases for condition-2, 52 cases for condition-3 and the rest 52 cases belonging to condition-4. In the Following, 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.

2) Simulation Data Sets

A Simulation Data Set (SDS) containing seven subsets is generated from the simulation system (SIMPACK and Track Spectrum). The SDS80, SDS120, SDS140, SDS160 and SDS200 correspond to 80 km/h, 120 km/h, 140 km/h, 160 km/h and 200 km/h, respectively. The experimental environment is described as follows, e.g., simulative locations: the corbel (car-body forepart); sampling frequency: 243Hz; mock conditions: seven conditions described in Table III. Each simulation data subset is a vibration signal. We divide the vibration signals into short frames of 280 at set intervals. A SDS has 14,000 cases: 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.

TABLE III
A DESCRIPTION OF THE SEVEN CONDITIONS

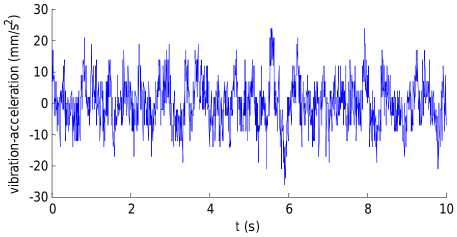
Type	**Condition**
condition-1	the normal train
condition-2	without anti-yaw shock absorber
condition-3	air spring failure
condition-4	without transverse shock absorber
condition-5	compound fault (condition-1, 2)
condition-6	compound fault (condition-1, 3)
condition-7	compound fault (condition-2, 3)

B. *Experimental Results*

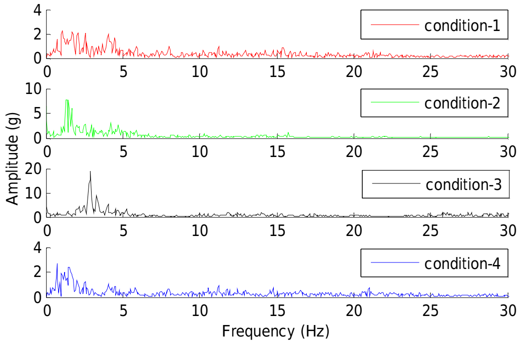
All experiments are performed with RDS for a total of 208 cases and SDS for a total of 14,000 cases unless otherwise stated. The experiments are as follows.

1) Experiment 1: Time and Frequency Domain Analysis

In analysis of time domain of the original signal (e.g., the speed of train is 120 km/h and the sampling time is 10 seconds), as shown in Fig. 4(a), the vibration signal is non-stationary. 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. 4(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.



(a)

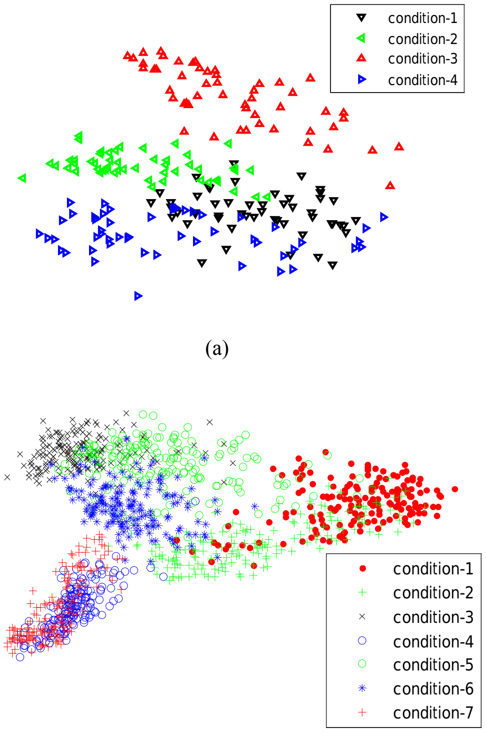


(b)

Fig. 4. The original signal: (a) time domain (b) frequency spectrum

2) Experiment 2: Learning Deep Features

In this experiment, the RDS160 and SDS160 selected from RDS and SDS are applied to observe the performance of our method in learning deep features from HST vibration signals. We train respectively (280)-100-100-60-30-2 nets (using a stack of four RBMs: 100-100, 100-60, 60-30, 30-2) on the RDS160 and SDS160 with 2 linear units in the "*top*" layer. For each RBM, we use 100 epochs to train it with a fixed learning rate of 0.1. In [25], Hinton gave a detailed explanation of the learning rate and momentum and sensible ways to set them. We show the 2-Dimensional representations by a (280)-100-100-60-30-2 net on the RDS160 in Fig. 5(a) and an alternative visualization of the 2-Dimensional representations by a (280)-100-100-60-30-2 net on the SDS160 in Fig. 5(b).



(a)

(b)

Fig. 5. (a) 2-Dimensional representations for 52 cases of each fault found by a (280)-100-100-60-30-2 net (b) an alternative visualization of the 2-Dimensional representations produced by a (280)-100-100-60-30-2 net trained on all 14,000 cases. 1,190 cases (170 per class) are sampled in a random order.

3) Experiment 3: Fault Recognition 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 [25]. We want to explore the performance of the DBNs in the fault recognition by varying the number of layers and the size of hidden units. For simplicity, we use the same size for every hidden layer in a network. All DBNs are pre-trained with a fixed method using a stochastic gradient decent with a mini-batch size of 1,260 training cases. For each RBM, we use 50 epochs to train it with a fixed learning rate of 0.1. According to Fig. 2, seven output units are connected to the "*top*" layer and the whole networks are fine-tuned. After several epochs of fine-tuning, the accuracy rate on the training data reaches a pre-specified threshold value (90%) and then the fine-tuning is stopped. The test accuracy rates at that point are shown in Figs. 6 and 7.

Fig. 6 shows the fault recognition accuracy rates for different numbers of layers. The noticeable trend in Fig. 6 is that a DBN with four layers shows a good performance. Fig. 7 shows the fault recognition accuracy rates for different sizes of each hidden layer. Overall, as shown in the Fig. 7, the recognition rate remains fairly constant with the increasing of the size of hidden units in each layer.

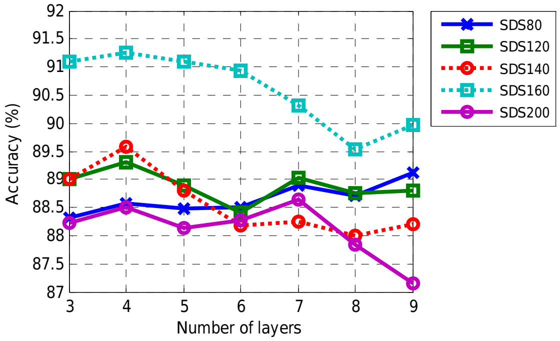


Fig. 6. Fault accuracy rate on the SDS on the basis of the number of layers, using 100 hidden units per hidden layer.

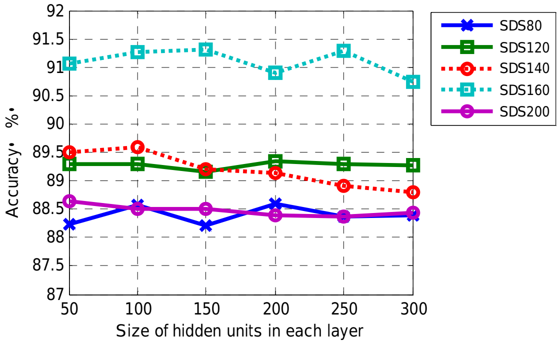


Fig. 7. Fault accuracy rate on the SDS on the basis of the size of hidden units in each layer, totally two hidden layers.

4) Experiment 4: Simulating the Real Environment

In order to test the performance of fault identification in simulating the real environment, the SDS consisting of seven faults with 14,000 cases is used for model building and the RDS consisting of four faults with 208 cases is used as the test set. We pre-train (280)-100-100-100 nets on SDS (SDS80, SDS120, SDS140, SDS160 and SDS200) and use 100 epochs to train each RBM with a fixed learning rate of 0.1. According to Fig. 2, seven output units are connected to the "*top*" layer and 30 epochs of discriminative fine-tuning of the whole networks. For comparison, BP neural network with three layers and SVM are also employed to analyze the same data sets, respectively. The parameters in BP are: "*hidden units = 50, the learning rate = 0.01, momentum = 0.9, mse (mean square error) = 0.001*". The settings of SVM (LSSVM Toolbox) are: "*model = initlssvm(traingin, trainingout, 'classifier', [], [], 'RBF_kernel'); model = tunelssvm(model, 'simplex', 'leaveoneoutlssvm', {'misclass'}, 'code_MOC'); model = trainlssvm(model)*".

The results of the three methods on the identification of the four different faults are listed in Table IV and shown in Fig. 8, respectively. From Table IV, it is noted that the FFT-DBNs model established on the simulation data sets is what exhibits the best recognition accuracy, 54.33% on the RDS200. It is also noted that the overall performance of the models for failure recognition is not high. And from Table IV and Fig. 8, it is clear that our method based on FFT-DBNs achieves the best identification results in the case of varying speeds. This comparison results imply that the presented method outperforms the other two methods in simulating the real environment.

TABLE IV
RECOGNITION ACCURACY IN SIMULATING THE REAL ENVIRONMENT (%)

Model	*v* (km/h)				
	80	*120*	*140*	*160*	*200*
FFT-BP	11.25	14.81	15.06	28.06	29.53
FFT-SVM	12.62	20.77	33.17	35.34	40.48
FFT-DBNs	**22.60**	**21.63**	**36.06**	**39.42**	**54.33**

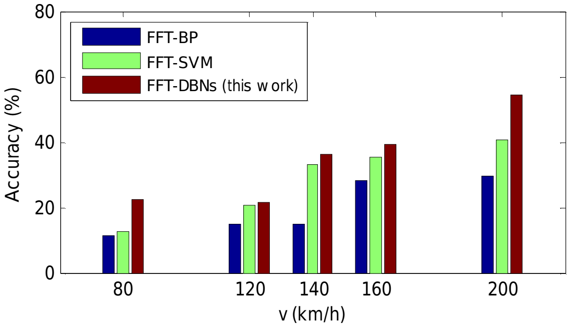


Fig. 8. Recognition accuracy of FFT-BP, FFT-SVM and FFT-DBNs.

IV. CONCLUSION

In this paper, we explored DBNs in circumstances of HST vibration. Vibration signals were analyzed in time and frequency domain in experiment 1 and the frequency range of the signals mainly concentrates on the low frequency band. The effectiveness for the DBNs on FFT to learn features from vibration signals was also investigated in experiment 2. The parameters of DBNs, the number of layers and the size of hidden units were explored in experiment 3. A DBN with four layers showed a good performance on SDS for a total of 14,000 cases and the recognition rate remained fairly constant with the increasing of the size of hidden units in each layer. Furthermore, the experimental results in simulating the real environment demonstrated higher classification accuracy than the others (e.g., FFT-BP and FFT-SVM). In our future work, we will focus on the investigation of high-level features over more fault data using the unsupervised DBNs method.

REFERENCES

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

[2] Lei Y, et al. EEMD method and WNN for fault diagnosis of locomotive roller bearings. *Expert Systems with Applications*, vol. 38, pp. 7334-7341, 2011.

[3] Chen L, Zi Y, Cheng W, et al. EEMD-1.5 Dimension spectrum applied to locomotive gear fault diagnosis. *International Conference on Measuring Technology and Mechatronics Automation*, vol. 1, pp. 622-625, 2009.

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

[5] Dash A K, Parhi D R. Analysis of an Intelligent Hybrid System for Fault Diagnosis in Cracked Structure. *Arabian Journal for Science and Engineering*, pp. 1-21, 2013.

[6] Raveendran R K S, Azariana M H, Kimb N H, et al. Effect of Multiple Faults and Fault Severity on Gearbox Fault Detection in a Wind Turbine using Electrical Current Signals. *Chemical Engineering Transactions*, vol. 33, pp. 79-84, 2013.

[7] Zhao J J, Yang Y, et al. Application of Empirical Mode Decomposition and Fuzzy Entropy to HST Fault Diagnosis. *The 8th International Conference on Intelligent Systems and Knowledge Engineering*, accepted, 2013.

[8] Li Z C. A Simple SOM Neural Network Based Fault Detection Model for Fault Diagnosis of Rolling Bearings. *Applied Mechanics and Materials*, vol. 397, pp. 1321-1325, 2013.

[9] Qin N, Jin W D, Huang J, et al. HST Bogie Fault Signal Analysis Based on Wavelet Entropy Feature. *Advanced Materials Research*, vol. 753, pp. 2286-2289, 2013.

[10] Zhang X L. Learning deep representation without parameter inference for nonlinear dimensionality reduction. *arXiv preprint arXiv: 1308.4922*, 2013.

[11] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks. *Science*, vol. 313, pp. 504-507, 2006.

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

[13] Lee H, Grosse R, Ranganath R, et al. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 609-616, 2009.

[14] Mohamed A, Dahl G E, Hinton G. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 14-22, 2012.

[15] Nair V, et al. 3-d object recognition with deep belief nets. *Advances in Neural Information Processing Systems*, vol. 22, pp. 1339-1347, 2009.

[16] Mnih V, et al. Learning to detect roads in high-resolution aerial images. *Computer Vision – ECCV 2010*, vol. 6316, pp. 210-223, 2010.

[17] Salakhutdinov R, Hinton G. Semantic hashing. *Int. J. Approximate Reasoning*, vol. 50, pp. 969-978, 2009.

[18] Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. *International Joint Conference on Artificial Intelligence*, vol. 14, pp. 1137-1145, 1995.

[19] Smolensky P. Information processing in dynamical systems: Foundations of harmony theory. *MIT Press*, 1986.

[20] Carreira-Perpinan M A, Hinton G E. On Contrastive Divergence Learning. *Artificial Intelligence and Statistics*, vol. pp. 17-24, 2005.

[21] Welling M, Hinton G E. A new learning algorithm for mean field Boltzmann machines. *Artificial Neural Networks—ICANN 2002*, vol. 2415, pp. 351-357, 2002.

[22] Tieleman T. Training restricted Boltzmann machines using approximations to the likelihood gradient. *Proceedings of the 25th international conference on Machine learning*, pp. 1064-1071, 2008.

[23] Bengio Y, Lamblin P, Popovici D, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, vol. 19, pp. 153-160, 2007.

[24] Salakhutdinov R. Learning deep generative models. *Diss. University of Toronto*, 2009.

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

[26] Bengio Y. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, vol. 2, pp. 1-127, 2009.