IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 23, 2017, accepted June 14, 2017, date of publication June 23, 2017, date of current version March 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2717492

Bearing Fault Diagnosis Using Fully-Connected Winner-Take-All Autoencoder

CHUANHAO LI[1], WEI ZHANG[1], GAOLIANG PENG[1], AND SHAOHUI LIU[2], (Member, IEEE)

[1]State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China
[2]Department of Computer Science, Harbin Institute of Technology, Harbin 150001, China

Corresponding author: Gaoliang Peng (pgl7782@hit.edu.cn)

This work was supported in part by the National High-Tech Research and Development Program of China (863 Program) under Grant 2015AA042201, in part by the National Natural Science Foundation of China under Grant 51275119, and in part by the Self-Planned Task of the State Key Laboratory of Robotics and System under Grant SKLRS201708A.

ABSTRACT Intelligent fault diagnosis of bearings has been a heated research topic in the prognosis and health management of rotary machinery systems, due to the increasing amount of available data collected by sensors. This has given rise to more and more business desire to apply data-driven methods for health monitoring of machines. In recent years, various deep learning algorithms have been adapted to this field, including multi-layer perceptrons, autoencoders, convolutional neural networks, and so on. Among these methods, autoencoder is of particular interest for us because of its simple structure and its ability to learn useful features from data in an unsupervised fashion. Previous studies have exploited the use of autoencoders, such as denoising autoencoder, sparsity aotoencoder, and so on, either with one layer or with several layers stacked together, and they have achieved success to certain extent. In this paper, a bearing fault diagnosis method based on fully-connected winner-take-all autoencoder is proposed. The model explicitly imposes lifetime sparsity on the encoded features by keeping only $k\%$ largest activations of each neuron across all samples in a mini-batch. A soft voting method is implemented to aggregate prediction results of signal segments sliced by a sliding window to increase accuracy and stability. A simulated data set is generated by adding white Gaussian noise to original signals to test the diagnosis performance under noisy environment. To evaluate the performance of the proposed method, we compare our methods with some state-of-the-art bearing fault diagnosis methods. The experiments result show that, with a simple two-layer network, the proposed method is not only capable of diagnosing with high precision under normal conditions, but also has better robustness to noise than some deeper and more complex models.

INDEX TERMS Autoencoder, fault diagnosis, lifetime sparsity, signal processing, signal representations, supervised learning, vibrations.

I. INTRODUCTION

With the rapid development of Industrial Internet, access to large amount of sensor data collected from various machines have become a reality. The availability of sensor data containing the information about the health condition of machines has given rise to the increasing business desire to resort to data-driven condition monitoring and fault diagnosis of machines [1]. Among different kinds of mechanical components, rolling element bearings are considered to be the key components in rotating mechanism. The health conditions of bearings have a direct and huge influence on the performance, stability and overall health conditions of the machine. The most common way to avoid possible damage is to monitor the health condition of rotating mechanism in real time through analyzing its vibration signals.

Data-driven condition monitoring and fault diagnosis techniques are usually based on machine learning algorithms. In recent years, a branch of machine learning algorithm based on neural networks, which is now rebranded as ''Deep Learning'', has achieved huge success in Computer Vision [2], [3] and Speech Recognition [4], [5]. Some deep learning techniques have already been applied to the field of machine health monitoring. Lu *et al.* [6] proposed a method based on stacked denoising autoencoders with three hidden layers for rotary machinery components fault diagnosis, using raw temporal input signal. They explored the influences of

receptive input size, number of hidden layers and nodes, sparsity constraint and the destruction level on the diagnosis accuracy. Some researchers use autoencoder models with frequency domain features as inputs instead of raw signals. Jia et al. fed the frequency spectra of time-series data into stacked denoising autoencoder (SAE) for rotating machinery diagnosis [7]. Huijie *et al.* [8] proposed a DNN model for hydraulic pump fault diagnosis that uses frequency features generated by Fourier transform as input. Liu *et al.* uses normalized spectrum generated by STFT of sound signal as inputs of a 2-layer DNN with pretraining process. Some researchers [9], [10] concatenate multi-domain statistical features including time domain, frequency domain and time-frequency domain features, and feed them into SAE as inputs. In [11], we proposed an end-to-end CNN model with 5 convolutional layers called WDCNN, which stresses the problem of noisy input and domain adaptation.

In real world applications, noise is an evitable problem. Traditional methods address this problem by denoising preprocessing, but this could slow down the speed of online diagnosing since all newly acquired signals need to be preprocessed first. Therefore, a model that can automatically learn noise-invariant features is desired. Whether model trained with samples collected under experiment conditions with few noise can achieve high accuracy when testing on noisy samples is an important issue [12], [13]. In Section IV-D of this paper, it is shown that some of the state-of-the-art DNN [7] fails to diagnose properly on noisy signals. Therefore, improvements on the robustness to noise can still be made.

In this paper, we proposed a method based on Fully-Connected Winner-take-all (FC-WTA) autoencoder [14], which explicitly impose lifetime sparsity on the encoded features, and combined with other constraints to learn extremely sparse and noise-robust features for the classification of bearing fault categories. An ensemble method is implemented to further enhance the performance of our proposed method. In the experiments, we show that our method has achieved better performance than the reference algorithms, and also has very good robustness against noise.

The structure of this paper is organized as follows. A brief introduction of autoencoder and some of its variations is provided in Section II. The intelligent diagnosis method based on FC-WTA autoencoder and the ensemble method used is introduced in Section III. In Section IV, experiments are conducted to explore the effects of some hyper-parameters and to compare the proposed method against some state-of-the-art methods. After this, visualization of the proposed model is presented to explore the inner mechanism of the proposed method. We draw the conclusions in Section V.

II. FULLY-CONNECTED WINNER-TAKE-ALL AUTOENCODER

In this section, we will give an introduction to general form of an autoencoder and different properties that we can impose on the features extracted by autoencoders. Then we will give a brief introduction about the FC-WTA autoencoders, and the modifications we have made on FC-WTA autoencoder to make it extract features with properties that we desire in bearing fault diagnosis.

A. AUTOENCODERS

Autoencoders were once used for layer-wise pre-training for deep convolutional neural networks [15]. However, it quickly fell out of fashion with the emergence of batch normalization [16], and residual learning [17]. With the new techniques, training deep networks from scratch is no longer a problem. However, autoencoders are still useful for its ability to learn useful information in the data and extracted features with some desired properties in an unsupervised way. Two of such properties useful for bearing fault diagnosis will be discussed in the subsections.

Autoencoder is a type of unsupervised algorithm that tries to learn an approximation to the identity function, so as to reconstruct the given input. A typical autoencoder usually consists of three parts: an encoder, a decoder, and a function that evaluates the information loss between input and output. Suppose we have an input x, an encoder function denoted as $h = f(x)$ which maps the input to encoded features, and a decoder function $y = g(h)$ that reconstruct the input signal from the encoded features. Then the parameters are adjusted to minimize the reconstruction error $L(x, y)$ which is usually the mean squared error (MSE) between input and its reconstruction. The general structure of an autoencoder is shown in Fig. 1.

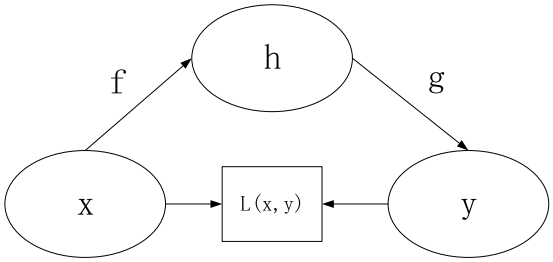


FIGURE 1. General Structure of an Autoencoder.

In [18], overcomplete autoencoders are discussed. In an overcomplete case where the hidden dimension is equal or greater than input dimension, the encoder and decoder can easily learn an identity function which will result in features that are not suitable for classification. Therefore, overcomplete autoencoders need to be regularized so that they can learn features with useful properties about the data distribution. In this section, we will talk about two useful properties that we are going to apply to the proposed method.

1) ROBUSTNESS TO NOISE

Denoising autoencoder (DAE) is first proposed by Bengio in 2008 [19]. The idea of denoising autoencoder (DAE) is to find a robust representation of the data by taking a partially corrupted data as input whilst training to reconstruct the

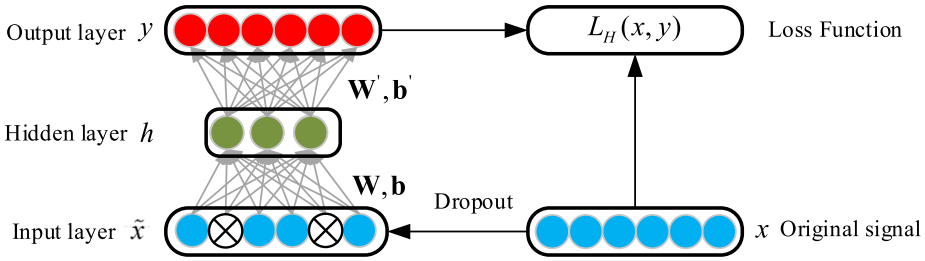


FIGURE 2. Structure of Denoising Autoencoder.

original undistorted input. To be a little more specific, in a typical DAE, some of the activations in input layer are randomly set to zero as a means to produce corrupted data (also called dropout), and then DAE tries to reconstruct the uncorrupted data from the corrupted ones.

Fig. 2 illustrates the structure of DAE, where x denotes the uncorrupted input, then it is corrupted to $\tilde{x}$. This is done by a stochastic mapping $\tilde{x} \sim q_D(\tilde{x}|x)$. Then the corrupted input $\tilde{x}$ is mapped to a hidden representation $\mathrm{h} = f(\tilde{x}) = f(W\tilde{x} + \mathrm{b})$. After this, DAE tries to reconstruct the uncorrupted input $\mathrm{y} = g(W'\mathrm{h} + \mathrm{b}')$ from h. During the training process, the parameters are adjusted to minimize the average reconstruction error $L_H(x, y) = H(B_x||B_y)$, whose purpose is to make y as close to x as possible.

After training the DAE, we can get a good representation of the original data, which is very robust to noise, since during the training process, by partially corrupting the inputs, DAE is forced to capture implicit invariances in the data. Robustness against noise is a very welcomed property when diagnosing vibration signals which are often influenced by noise in real-world working conditions.

2) SPARSITY OF THE REPRESENTATION

As discussed in [20], the advantages of sparse features include information disentangling, efficient variable-size representation, linear separability, and being distributed but sparse. Therefore, sparsity is usually a desired property for classification.

An autoencoder whose optimization objective contains a penalty term $\Omega(\mathrm{h})$ on sparsity level of the encoded features h is usually called a sparse autoencoder. Therefore, now the autoencoder tries to minimize the following Eq. (1) instead of only the reconstruction error:

$$L_{sparse} = L(x, y) + \Omega(\mathrm{h}) \tag{1}$$

A common way to implement this sparsity penalty is to use the KL divergence between the hidden unit marginal $\hat{\rho}$ and the target sparsity probability ρ. Let $h_j(x)$ denotes the activation of hidden unit j when the network is given a specific input x. Then the average activation of hidden unit j over whole training set is:

$$\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}[h_j(x^i)] \tag{2}$$

where m denotes the number of training samples. Now the KL divergence can be written as:

$$\sum_{j=1}^{n_2}\mathrm{KL}(\rho\|\hat{\rho}_j) = \sum_{j=1}^{n_2}\rho\log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j} \tag{3}$$

where n_2 denotes the number of neurons in the hidden layer. So our new optimization objective is:

$$L_{sparse} = L(x, y) + \lambda\sum_{j=1}^{n_2}\mathrm{KL}(\rho\|\hat{\rho}_j) \tag{4}$$

where $L(x, y)$ is the reconstruction error that we have already defined, and λ controls the weight of the sparsity penalty term.

The above sparse autoencoder places a "lifetime sparsity" on the encoded features to ensure that each neuron in the hidden layer will only be activated for small number of samples. However, this approach only works for sigmoidal autoencoders and it's hard to choose the right λ parameter. However, fully-connected winner-take-all autoencoder can be used to cope with these problems which also places lifetime sparsity on encoded features.

B. FULLY-CONNECTED WINNER-TAKE-ALL AUTOENCODER

Fully-Connected Winner-take-all (FC-WTA) autoencoder was proposed to address the problems faced by the sparse autoencoder which uses KL divergence as penalty. It can be used on ReLU autoencoders and has only one main hyper-parameter that needs to be tuned (the sparsity rate).

In the feedforward stage, the input x is encoded into h by the encoder, which can be one or more than one ReLU layers. However, different from an ordinary autoencer which would directly use this encoded feature h to reconstruct x, FC-WTA need to impose a lifetime sparsity by keeping only the k% largest activations of each hidden unit across whole mini-batch, and set the rest to zero. And during test time, the sparsity constraint is turned off, and the final representation is the output of the encoder.

In this paper, the FC-WTA proposed by Makhzani and Frey [14] is modified by randomly corrupting input signal to gain robustness against noise as discussed in 2.1.1, adding other penalty terms including $l1$ penalty on activations of encoded features after lifetime sparsity is imposed, and

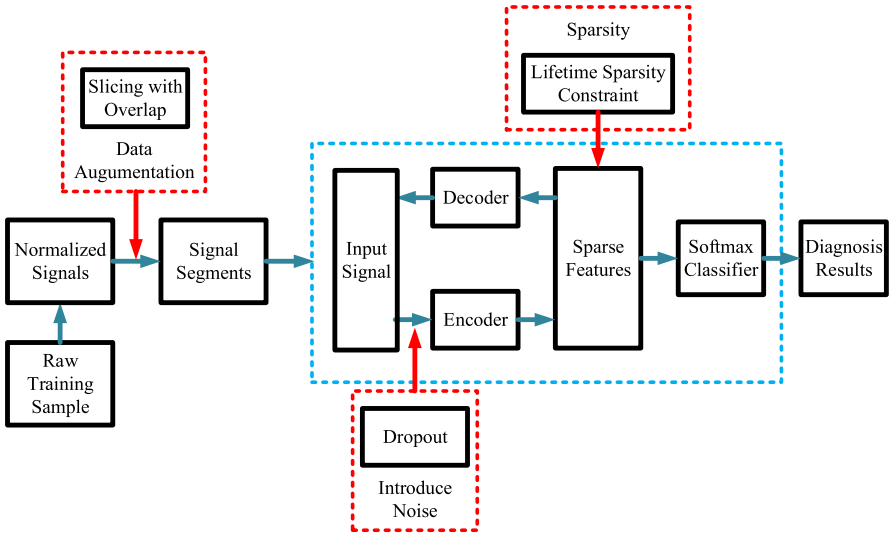


FIGURE 3. Training process of proposed method.

also $l2$ penalty on weights of the autoencoder to prevent overfitting. Therefore, the loss function is:

$$L_{FC-WTA}(x, y) = L(x, y) + \alpha \times \frac{1}{n_2} \sum_{i=1}^{n_2} |h_i(x)| + \frac{\lambda}{2} \sum_{w} w^2 \tag{5}$$

where λ denotes the weight of the $l2$ penalty, w is the weight of the autoencoder, α denotes the weight of $l1$ penalty on hidden activations $h_i(x)$, and $L(x, y)$ is the reconstruction loss.

III. UNSUPERVISED LEARNING, SUPERVISED FINE TUNING, AND ENSEMBLE

This section gives detailed information about the training and testing process of the proposed method for bearing fault diagnosis. The training process can be divided into two stages, namely, unsupervised feature learning with fully connected winner-take-all autoencoder and supervised fine tuning. In order to increase the performance and reduce variance of the classification results when testing, time and model ensemble methods are implemented. The flowchart of the training process of proposed method is shown in Fig. 3.

During the unsupervised feature learning stage, fully connected winner-take-all autoencoder is used to learn sparse features from the normalized vibration signals, and then follows the supervised fine tuning process which makes the encoder adjust its weights to extract features that are more suitable for fault diagnosis. During testing process, a sliding window is applied on sample signal to slice multiple signal sequences, and then feed them into the fully connected winner-take-all autoencoder for classification. The softmax output of each segments are averaged as a means of soft voting, which would effectively improve the classification accuracy.

A. UNSUPERVISED FEATURE LEARNING USING UNLABELED DATA

The unsupervised feature learning stage consists of three steps as can be seen in Fig. 4. We first randomly drop some points of the input signal as means to introduce noise. Then the signal is mapped to hidden layer, which has a lifetime sparsity constraint that keeps only the k% largest activations while setting the rest to zero. Then the network tries to reconstruct the uncorrupted input signal from the sparsified hidden units. In this way, the winner-take-all autoencoder will be forced to learn features that are sparse and robust to noise.

Let's use N_{in} to denote input dimension of the fully connected WTA autoencoder, and $N_{features}$ the dimension of the learned sparse features. A simple data augmentation technique is used to increase the number of training samples, that is, slicing the training samples with overlap. The process is shown in Fig. 5. The training samples is prepared with overlap. For example, a vibration signal with l_{signal} points can provide $N_{samples} = \frac{l_{signal} - N_{in}}{s}$ training samples when each segment has length N_{in}, and the stride of sliding window is s.

Then we apply min-max normalization method to normalize each segment into the [0, 1] interval. The equation to apply min-max normalization is:

$$\tilde{\mathbf{x}}^i = \frac{\mathbf{x}^i - \min(\mathbf{x}^i)}{\max(\mathbf{x}^i) - \min(\mathbf{x}^i)} \tag{6}$$

where $\mathbf{x}^i \in \mathbb{R}^{N_{in}}$ denotes the i-th training sample which contains N_{in} data points, $\min(\cdot)$ and $\max(\cdot)$ denote the operations that return the minimum and maximum value in data segment x^i, respectively.

After the normalized signal segment is fed into the network, it first needs to be randomly corrupted to introduce noise, so that the autoencoder will later learns to reconstruct uncorrupted signal from the corrupted input, which will

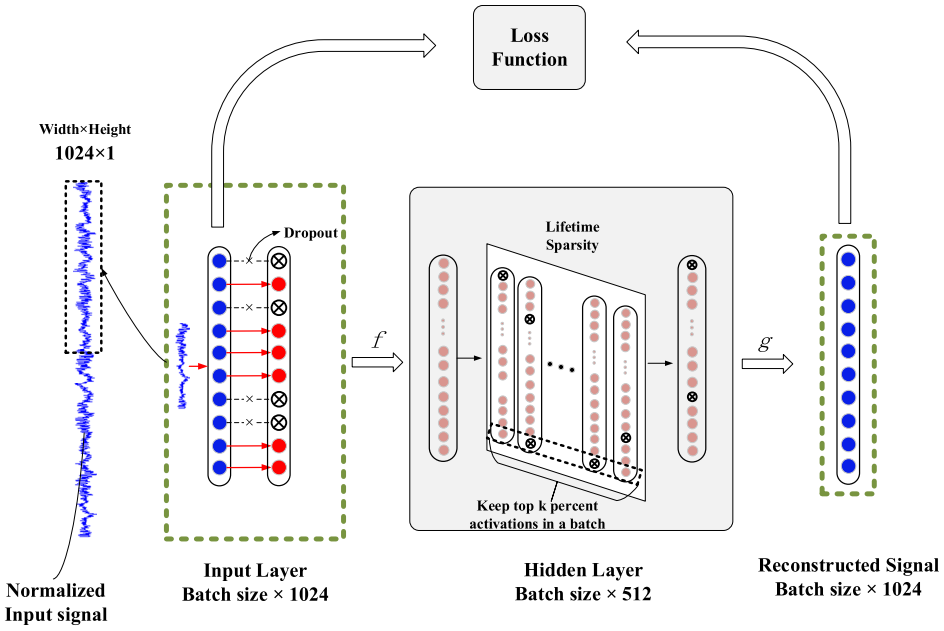


FIGURE 4. Unsupervised feature learning with fully-connected WTA autoencoder.

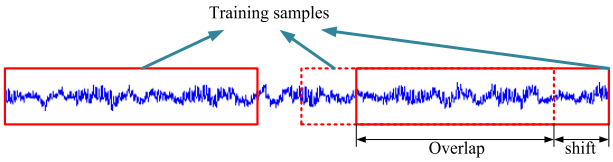


FIGURE 5. Data augmentation by slicing with overlap.

enhance its robustness to noise. This is done by a stochastic mapping $\tilde{x}^i \sim q_D(\tilde{\mathbf{x}}^i|\mathbf{x}^i)$. This process is illustrated in the left part of Fig. X.

The encoder consists of one or more ReLU layers, which is denoted as $f(\cdot)$. Thus, the hidden layer $\mathbf{h}$ is computed as:

$$\mathbf{h}^i = f(\tilde{\mathbf{x}}^i) \tag{7}$$

where $\mathbf{h}^i \in \mathbb{R}^{N_{features}}$ denotes the hidden layer of the i-th training samples. Then we impose a lifetime sparsity by keeping the largest $k\%$ activation of each hidden unit across all the samples in a mini-batch, and setting the rest of activations of each hidden unit in this mini-batch to zero. Algorithm 1 shows the steps to impose lifetime sparsity.

After imposing lifetime sparsity on hidden layer, we use the decoder function $g(\cdot)$ to map the hidden layer $\hat{h}^i \in \mathbb{R}^{N_{features}}$ to output $\mathbf{o}^i \in \mathbb{R}^{N_{in}}$:

$$\mathbf{o}^i = g(\hat{\mathbf{h}}^i) \tag{8}$$

where $g(\cdot)$ is usually composed of one Softplus layer.

The FC-WTA autoencoder then needs to adjust the weights and biases of the layers in the encoder $f(\cdot)$ and decoder $g(\cdot)$ to minimize the optimization objective as described by Eq. (5) in Section II-B, which includes the mean squared error between uncorrupted input $\mathbf{x}^i$ and network output $\mathbf{o}^i$, the $l1$ penalty on hidden layer activation $\hat{\mathbf{h}}^i$ to impose population sparsity, and

Algorithm 1 Lifetime Sparsity on Hidden Layer

Input: Hidden layer activations of all the samples in a mini-batch $\mathbf{H} \in \mathbb{R}^{N_{features} \times N}$:

$$\mathbf{H} = \{\mathbf{h}^1, \mathbf{h}^2, \mathbf{h}^3, \mathbf{h}^4, \cdots, \mathbf{h}^i, \cdots, \mathbf{h}^N\}$$

where $\mathbf{h}^i \in \mathbb{R}^{N_{features}}$ denotes the hidden layer of the i-th training samples, and N denotes the batch size

Lifetime sparsity rate k%

Output: Hidden layer activations with lifetime sparsity imposed

$$\hat{\mathbf{H}} = \{\hat{\mathbf{h}}^1, \hat{\mathbf{h}}^2, \hat{\mathbf{h}}^3, \hat{\mathbf{h}}^4, \cdots, \hat{\mathbf{h}}^i, \cdots, \hat{\mathbf{h}}^N\}$$

For Each hidden neuron a_j in the hidden layer, where $j = (1, 2, 3, \cdots, N_{features})$

Find the top k% × N activation values of a_j across the mini-batch $a_j^1, a_j^2, a_j^3, \cdots, a_j^i, \cdots, a_j^N$, which is denoted by top_values $\in \mathbb{R}^{k\% \times N}$

$$\hat{a}_j^i = \begin{cases} a_j^i & if\ a_j^i \geq \min(\text{top\_values}) \\ 0 & if\ a_j^i < \min(\text{top\_values}) \end{cases}$$

End for

$l2$ penalty on all the weights in encoder $f(\cdot)$ and decoder $g(\cdot)$ to prevent overfitting.

B. SUPERVISED FINE TUNING WITH LABELED DATA

After unsupervised feature learning process, the highly non-linear encoder function of FC-WTA autoencoder learns to do sparse coding. To improve its classification performance,

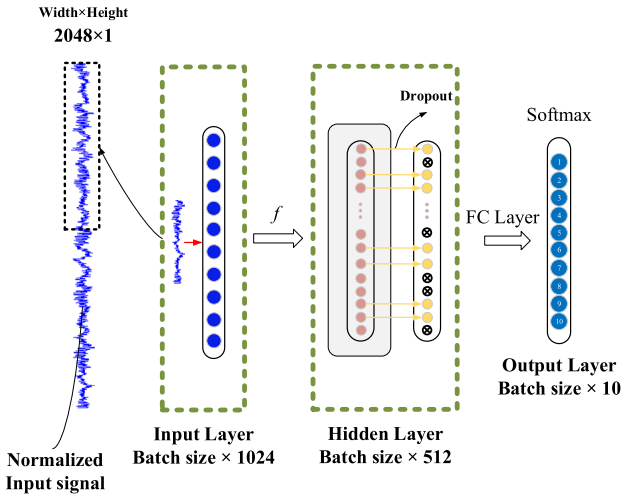


FIGURE 6. Supervised fine-tuning process.

we use labeled data to fine-tune the network. As is shown in Fig. 6, during the supervised fine tuning process, the lifetime sparsity constraint placed on the hidden units is relaxed, the dropout placed on input signals is also taken away.

Given labeled training samples $(\mathbf{x}^i, \mathbf{y}^i)$, where $\mathbf{x}^i \in \mathbb{R}^{N_{in}}$, and $\mathbf{y}^i \in \mathbb{R}^{10}$, which is one-hot encoding of the bearing health condition categories. The encoding part of the autoencoder extracts sparse features $\mathbf{h}^i = f(\mathbf{x}^i)$ from uncorrupted input signal $\mathbf{x}^i$ with the encoder function $f(\cdot)$ trained in the unsupervised process. Then dropout is applied on the sparse features $\mathbf{h}^i$ as a regularization method. Finally, a softmax classifier is used to map the extracted sparse features $\mathbf{h}^i$ to the ten categories representing the predicted health conditions of the bearings, which is denoted by $\mathbf{p}^i \in \mathbb{R}^{10}$.

The optimization objective is the cross entropy loss between $\mathbf{y}^i$ and $\mathbf{p}^i$, which is shown in Eq. (9):

$$\mathrm{H}\left(\mathbf{y}^i, \mathbf{p}^i\right) = -\sum_{i=1}^{m} \mathbf{y}^i \log\left(\mathbf{p}^i\right) \tag{9}$$

where m denotes the number of training samples in a mini-batch.

C. ENSEMBLE METHOD BY SOFT VOTING AMONG SIGNAL SEGMENTS

In order to enhance the performance of the proposed method during test stage, an ensemble method is applied. To be more specific, for each test signal segment, several sub-segments are sliced from it with overlap, and then diagnosis is conducted on each of these sub-segments, whose results are combined to get the final diagnosis result via soft voting.

During test stage, each test signal of length 2048 (also normalized using min-max normalization) is sliced into five segments of length 1024. Then proposed method based on FC-WTA autoencoder is used to diagnose each segments. The outputs of sofmax classifier are five vectors that indicate the probability of each segment belonging to the bearing health condition categories. These five softmax outputs are used for soft voting. To be more specific, the softmax outputs for five input segments are averaged to get a final probability distribution over ten categories. Therefore, the final classification decision is simply the category that receives the highest probability. This process is shown in Fig. 7.

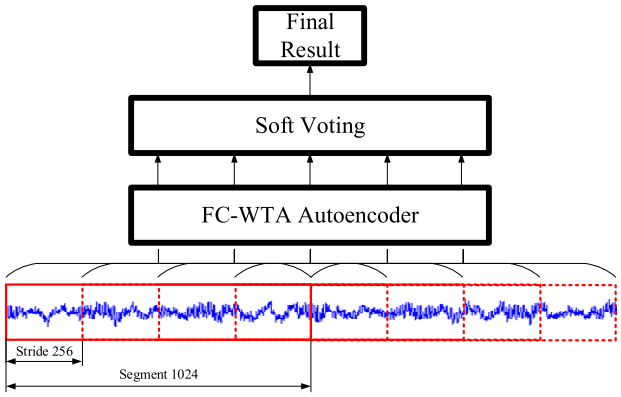


FIGURE 7. Soft voting among signal segments to aggregate five prediction results.

For each test signal we divide it into a set of five segments $\{s_1, s_2, s_3, s_4, s_5\}$, and these segments are fed into the classifier to predict the class label from a set of 10 possible class labels $\{c_1, c_2, c_3, \cdots, c_{10}\}$. For a segment s_i, the outputs of the classifier are given as a 10 dimensional label vector $\mathbf{p}^i = (\mathbf{p}^i_1, \mathbf{p}^i_2, \cdots, \mathbf{p}^i_j, \cdots, \mathbf{p}^i_{10})^T$, where $\mathbf{p}^i_j$ is the output of segment s_i for class label c_j. Therefore, soft voting method is simply averaging over $\mathbf{p}^i$ for $\mathrm{i} = 1, 2, \cdots, 5$.

$$\bar{\mathbf{p}} = \frac{1}{5}(\mathbf{p}^1 + \mathbf{p}^2 + \cdots + \mathbf{p}^5) \tag{10}$$

where $\bar{\mathbf{p}} \in \mathbb{R}^{10}$ indicates the probability distribution for each category that soft voting gives us.

IV. EXPERIMENTAL VERIFICATION OF PROPOSED METHOD

Rolling element bearings are key components in rotary mechanism, and their health conditions often have major impact on the performance and reliability of the mechanism they are part of. Therefore, in this section, the proposed method will be verified on the benchmark bearing dataset. In the experiment, the performance of proposed fully-connected winner-take-all autoencoder approach is compared with some state-of-art diagnosis algorithms, with the detailed results listed below. Also, considering noise is unavoidable in real world industrial production, as the vibration signals are easily contaminated by noise, the ability to diagnose fault types under the noisy environment is crucial and challenging. In order to test the performance of the proposed method on noisy signals, we generated a new dataset by adding white Gaussian noise to the original signals, trying to simulate the corrupted signals collected in real world applications. In the reminder of this section, we will investigate the performance of the proposed method under these two scenarios.

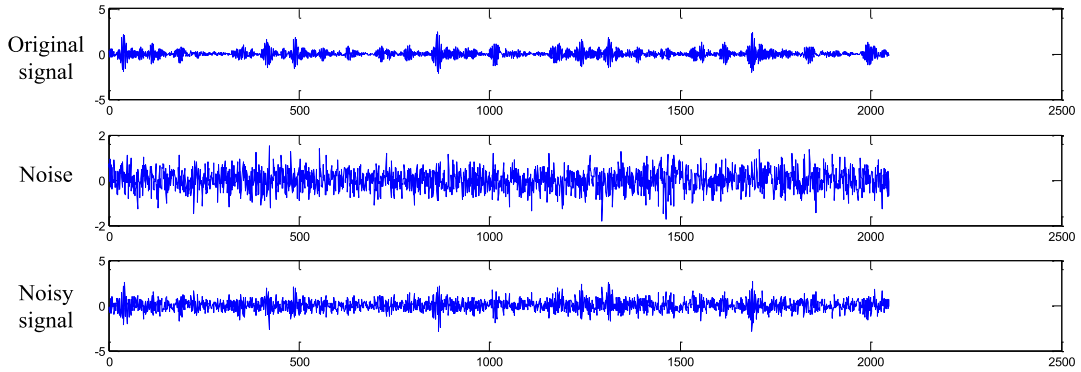


FIGURE 8. Illustrations of original signal, white Gaussian noise, and noisy signal.

TABLE 1. Description of rolling element bearing datasets.

Fault location		none	Ball			Inner race			Outer race			Load
Category Label		1	2	3	4	5	6	7	8	9	10	1,2,3
Fault diameter(mm)		0	0.007	0.014	0.021	0.007	0.014	0.021	0.007	0.014	0.021	
Dataset	Train	1980	1980	1980	1980	1980	1980	1980	1980	1980	1980	
Size	Test	75	75	75	75	75	75	75	75	75	75	

A. DATA DESCRIPTION

The dataset used for the verification of proposed method is from the Case Western Reserve University (CWRU) Bearing Data center [21]. The data was obtained from the accelerometers of the motor driving mechanical system at a sampling frequency of 12 kHz. In this dataset, there are four possible health conditions of the bearing: normal, ball fault, inner race fault and out race fault. Each fault type contains fault diameters of 0.007 inch, 0.014 inch and 0.021 inch respectively, so we have ten categories of health conditions in total. In this experiment, the train set contains 39600 training data, and each training sample contains 1024 data points. Test set contains 750 test samples and each has length 2048 so that we can use the majority voting method over sub-segments of each sample. Both train and test set contain signals collected under all three loads. In addition, the training samples are overlapped to augment data and there is no overlap among the test samples. The details of all the datasets are described in Table 1.

In addition, to evaluate the diagnosis accuracy of the proposed method on signals with white Gaussian noise, the model is trained on the original data from CWRU, then it will be tested on data with various amount of noise.

The noise dataset is generated in order to simulate the signals collected from real world scenarios where many different factors could introduce noise to the signal. For each original signal x, we first measure the power of x, then add white Gaussian noise of a specific signal-to-noise ratio (SNR) to signal x. The definition of SNR is shown below:

$$SNR_{dB} = 10\log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) \tag{11}$$

where P_{signal} and P_{noise} are the power of signal and the noise respectively.

Fig. 8 gives an example of the original signal, white Gaussian noise, and the signal added with white Gaussian noise. The SNR for the composite noisy signal is 0 dB, which means the power of noise is equal to that of the original signal. We test the proposed FC-WTA method on noisy signals whose SNR value ranges from -4dB to 10dB.

B. HYPER-PARAMETER SELECTION

The proposed method has some hyper-parameters that need to be decided to get optimal performance. Some experiments were undertaken to analyze the effect of the changes in such parameters in the FC-WTA autoencoder model setup. In this section, various configurations of sparsity rate and units, sparsity proportion, and data destruction level were tried to achieve the optimal performance.

In [6], reconstruction errors of the autoencoders are set to be the indicators to choose hyper-parameters, which may not be very proper. As stated in [18], if an autoencoder succeeds in simply learning an identity function, then it is not especially useful. Usually all kinds of constraints are imposed on autoencoders to force them to prioritize which aspects of the inputs should be copied, and this may enable them to learn useful features of the data. Therefore, reconstruction error is really not a good metric for finding optimal configurations of hyper-parameters. Instead, we should use the ultimate purpose of the method as means to decide what is a good configuration of hyper-parameters. In our paper, our main purpose is to configure our model to achieve highest accuracy in diagnosing signals under various noisy conditions, so

TABLE 2. Diagnosis accuracy on noisy signals with SNR = 10.

SNR = 10		Lifetime Sparsity Rate						
		0.01	0.05	0.08	0.1	0.15	0.2	1
L_1	0	96.29%	97.68%	97.76%	97.87%	98.48%	98.00%	93.97%
	0.005	96.29%	96.61%	97.73%	98.53%	98.45%	98.08%	95.01%
	0.01	96.45%	97.52%	98.13%	98.08%	98.19%	98.05%	95.17%
	0.02	96.59%	98.05%	97.89%	98.53%	98.11%	98.43%	94.80%
	0.05	96.88%	97.89%	97.65%	98.24%	97.84%	98.11%	93.09%
	0.1	96.77%	97.95%	97.31%	98.13%	97.92%	97.60%	94.83%
	1	96.96%	97.87%	97.89%	97.68%	96.40%	95.71%	79.52%

TABLE 3. Diagnosis accuracy on noisy signals with SNR = 6.

SNR = 6		Lifetime Sparsity Rate						
		0.01	0.05	0.08	0.1	0.15	0.2	1
L_1	0	93.81%	96.77%	96.19%	96.35%	97.41%	97.39%	91.39%
	0.005	93.39%	95.84%	96.51%	97.07%	97.47%	97.15%	92.43%
	0.01	93.87%	96.56%	96.77%	96.77%	97.12%	97.39%	92.24%
	0.02	94.24%	96.93%	96.56%	97.33%	97.12%	97.20%	92.05%
	0.05	94.72%	96.77%	96.67%	97.04%	96.96%	96.69%	90.05%
	0.1	93.49%	96.88%	96.32%	97.28%	97.01%	96.35%	92.27%
	1	94.56%	96.72%	96.72%	96.05%	94.21%	92.96%	73.44%

it's more reasonable to use diagnosis accuracy under noisy environment as criteria to determine hyper-parameters.

The hyper-parameters that may have major influence on the proposed method include number of hidden layers and nodes, sparsity rate and units, sparsity proportion, and data destruction level. In the proposed method, the number of ReLU layers in the encoder is set to be one, and the number of nodes in that ReLU layer is 512. In the reminder of this section, experiments are conducted to try out different settings of these hyper-parameters.

1) SPARSITY RATE AND WEIGHT OF L_1 PENALTY

Sparsity rate controls the lifetime sparsity of the encoded representations. To be more specific, it controls the percentage of activations of each neuron that can be kept across all samples in a mini-batch. And the weight of L_1 penalty on hidden activations controls both lifetime sparsity and population sparsity by penalizing all activations of the neurons in the last layer of encoder in a mini-batch. To explore the influences of sparsity on diagnosis accuracy, we vary the value of sparsity rate and weight of L_1 penalty to see the diagnosis accuracy on validation samples with white Gaussian noise. The results can be seen in Table 2 to 5.

As can be seen from Table 2-5 above, when testing on signals with white Gaussian noise whose SNR values are 10, 6, 2, -2, better diagnosis performances are achieved when the lifetime sparsity is 0.1 and 0.15. For the weight of L_1, there seems to be some differences between signals with higher SNR values and signals with lower SNR values. Better results are achieved when weight of L_1 is set to be 0.005 for signals whose SNR value is 10 and 6, while signals whose SNR value is 2 and -2 seem to prefer higher weight of L_1 like 0.05. However, there are not enough evidence to draw a reliable conclusion, but it is enough to provide some hints on the selection of lifetime sparsity rate and weight of L_1. Considering the choice of this hyper-parameter has smaller influence on the diagnosis of noise with less noise whose test accuracy is already pretty high, in the following experiments we will set lifetime sparsity rate to be 0.15 and weight of L_1 to be 0.05 to maximize the performance on signals with more noise.

2) DESTRUCTION LEVEL

The destruction level refers to the probability that each data point of input signals is dropped. Higher destruction level means introducing bigger noise to the input, and potentially this may increase the model's robustness to noise, because the model is forced to learn noise-invariant representations. However, excessive destruction level may lead to loss of useful information, and thus the model would fail to reconstruct the original properly. Fig. 9 illustrates the influence of destruction level on diagnosis accuracy of FC-WTA method. We changed the probability of each data point in the input signal to be kept from 0.1 to 1, and use a validation set with white Gaussian noise added to see the changes in performance. The results show that as long as the keeping probability is larger

TABLE 4. Diagnosis accuracy on noisy signals with SNR = 2.

SNR = 2		Lifetime Sparsity Rate						
		0.01	0.05	0.08	0.1	0.15	0.2	1
L_1	0	87.65%	93.97%	93.39%	93.60%	94.48%	94.48%	86.59%
	0.005	87.01%	92.45%	93.20%	93.23%	94.03%	93.47%	85.12%
	0.01	88.27%	94.32%	93.12%	94.16%	93.92%	93.73%	85.57%
	0.02	88.61%	93.73%	93.33%	94.27%	94.32%	92.43%	85.41%
	0.05	89.36%	93.41%	93.84%	93.33%	94.69%	92.99%	84.59%
	0.1	86.93%	93.09%	93.95%	94.29%	93.79%	92.96%	85.41%
	1	89.76%	92.93%	93.28%	91.52%	87.81%	84.83%	63.09%

TABLE 5. Diagnosis accuracy on noisy signals with SNR = −2.

SNR = -2		Lifetime Sparsity Rate						
		0.01	0.05	0.08	0.1	0.15	0.2	1
L_1	0	72.08%	80.45%	80.93%	80.00%	81.28%	81.84%	69.92%
	0.005	71.36%	79.47%	79.60%	79.15%	80.75%	79.55%	70.19%
	0.01	74.03%	83.15%	79.57%	83.09%	80.11%	81.44%	68.77%
	0.02	74.21%	81.55%	79.39%	81.81%	83.20%	78.11%	69.31%
	0.05	74.29%	81.87%	82.13%	80.83%	83.92%	78.99%	69.04%
	0.1	73.81%	79.87%	82.72%	81.55%	81.41%	78.69%	70.40%
	1	73.23%	79.47%	80.45%	77.09%	70.37%	67.33%	48.69%

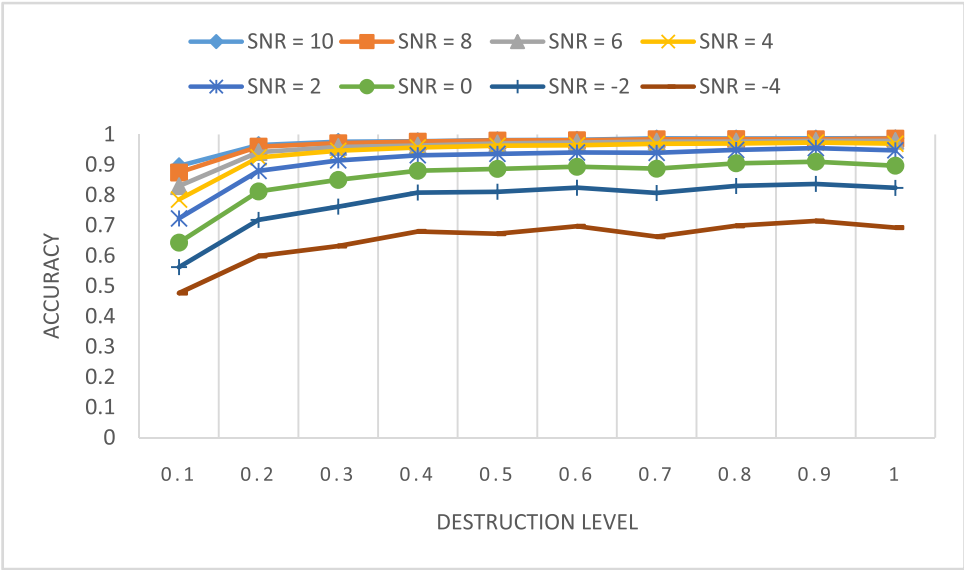


FIGURE 9. The Influence of Destruction Level on Diagnosis Accuracy of FC-WTA method on Noisy Signals with various SNR values.

than 0.5, the performance is good. However, this results also show that randomly corrupting input signals may not have an obvious impact on the performance of FC-WTA method, because we can see the performance with keeping probability being 1 is the same with that being bigger than 0.6, and too large dropout rate will even decrease the diagnosis accuracy.

C. THE INFLUENCE OF SOFT VOTING AMONG SIGNAL SEGMENTS

In Section IV-B, we explored the influence of different configurations of hyper-parameters on the diagnosis accuracy, and the proposed method can achieve pretty good results. To further enhance the accuracy and stability of the proposed

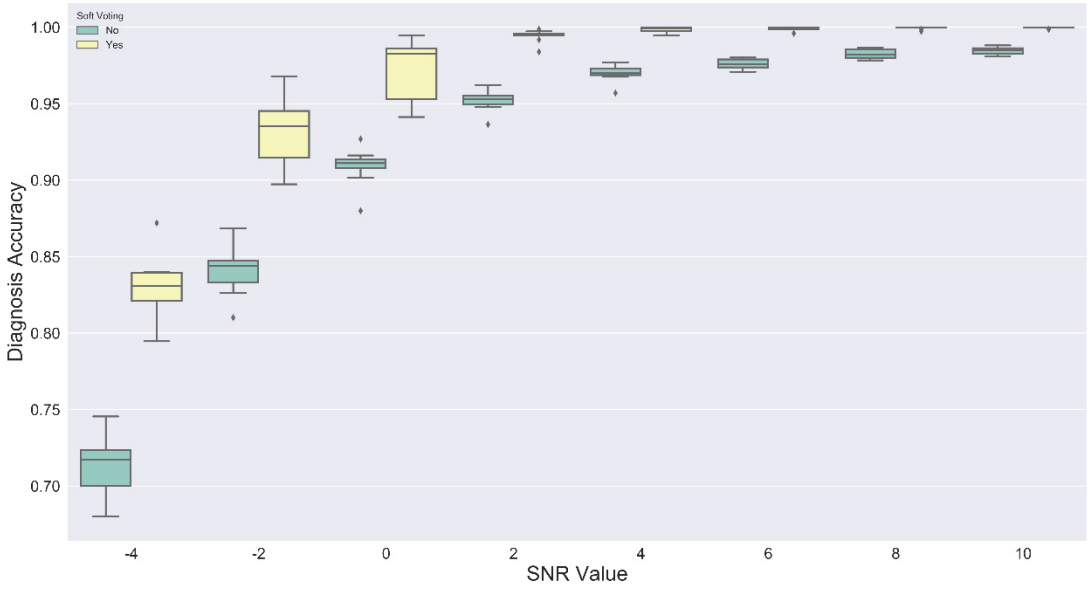


FIGURE 10. Improvement on diagnosis accuracy with soft voting method.

method, the ensemble method introduced in Section III-C will be implemented. In this set of experiments, we will apply soft voting method to aggregate the prediction results of 5 segments from proposed FC-WTA autoencder method. In the remainder of the paper, FC-WTA method with soft voting among signal segments will be denoted as FC-WTA Ensemble.

The improvement on stability and accuracy of FC-WTA method with soft voting is studied by repeatedly diagnosing the same validation set for ten times. The results are shown in Fig. 10. We can see that there is a very obvious improvement on the diagnosis accuracy across noisy signals with all SNR values. However, it also shows a decrease in the diagnosis stability on signals whose SNR is -4, -2 or 0, and increase in that on signals whose SNR is 2-10. This suggests than when there is too much noise, the classifier is not certain about its prediction and aggregating multiple predictions does not help to increase this certainty.

D. COMPARISON WITH REFERENCE ALGORITHMS ON NOISY SIGNALS

In this part, we are going to compare the performance of FC-WTA and FC-WTA Ensemble with that of some of the state-of-the-art methods, including SVM and DNN whose inputs are frequency features extracted by FFT, and WDCNN whose input is normalized temporal signal. The final configurations of the hyper parameters of the proposed method is shown in Table 6. The comparison results are shown in Fig. 11.

As can be seen from Fig. 11, the diagnosis accuracy of SVM, WDCNN, DNN, FC-WTA, and FC-WTA Ensemble with different SNR of the input signals are compared. Among the three reference algorithms, SVM and DNN are based on

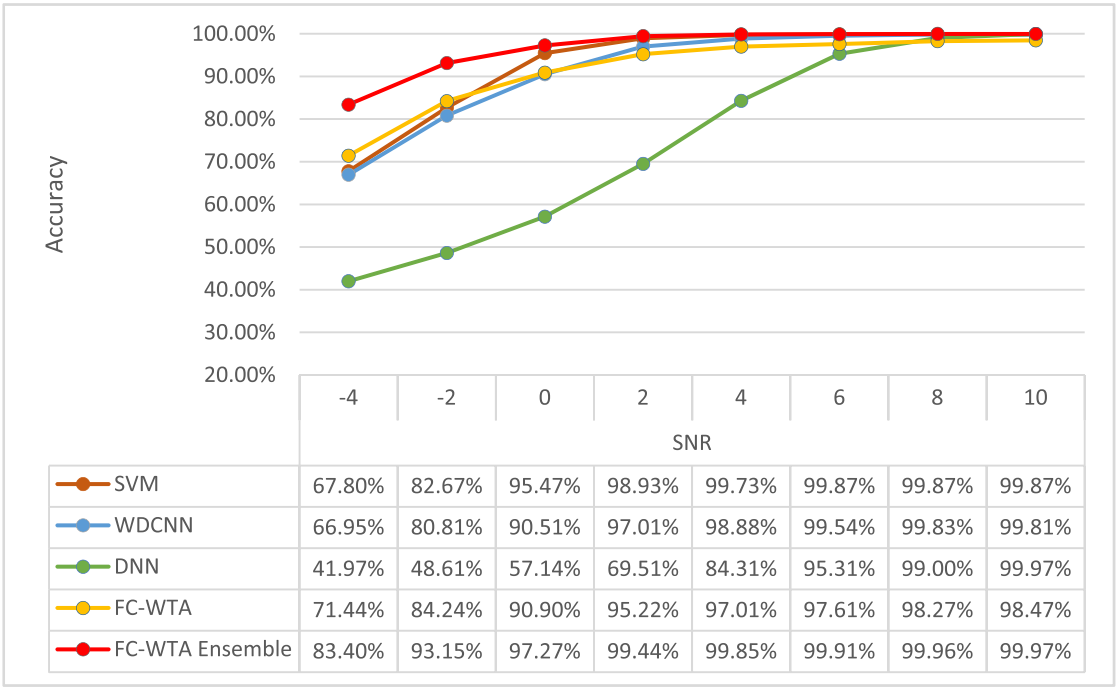
TABLE 6. Hyper parameters of FC-WTA ensemble method.

Hyper parameter	Setting
Input size	1024
Destruction level	10%
Number of ReLU layers in encoder	1
Number of nodes	512
Lifetime Sparsity rate	0.15
Weight of L_2 penalty	0.004
Weight of L_1 penalty	0.05

frequency features extracted by FFT. The kernel function of SVM is radial basis function. The number of neurons in each layer of MLP is 1025, 500, 200, and 10, respectively, and the activation function is Sigmoid. Fig. 11 illustrates that when testing on signals with small amount of noise (SNR value being 8 or 10) added, all five algorithms are capable of diagnosing at nearly 100% accuracy. However, with the decrease of SNR value, the accuracy of DNN decreases remarkably, while the other algorithms remain relatively stable. When SNR value of the test samples is 0 dB, the accuracy of DNN drops below 60%, and that of the proposed FC-WTA and FC-WTA Ensemble are still above 90%. When testing on extremely noisy samples with SNR being -4, the accuracies of SVM, WDCNN, and FC-WTA have decreased to about 70%, while that of FC-WTA Ensemble still remains about 80%. This suggests that compared with the traditional intelligent fault diagnosis method and the state of the art DNN method, FC-WTA Ensemble has much better robustness against noise.

E. VISUALIZATIONS OF THE NETWORK

Generally, it is hard to understand the inner operating mechanism of neural networks. In this section, we try to explore some of the inner mechanism of the proposed FC-WTA



	-4	-2	0	2	4	6	8	10
SVM	67.80%	82.67%	95.47%	98.93%	99.73%	99.87%	99.87%	99.87%
WDCNN	66.95%	80.81%	90.51%	97.01%	98.88%	99.54%	99.83%	99.81%
DNN	41.97%	48.61%	57.14%	69.51%	84.31%	95.31%	99.00%	99.97%
FC-WTA	71.44%	84.24%	90.90%	95.22%	97.01%	97.61%	98.27%	98.47%
FC-WTA Ensemble	83.40%	93.15%	97.27%	99.44%	99.85%	99.91%	99.96%	99.97%

FIGURE 11. Comparison among SVM, WDCNN, DNN, FC-WTA, and FC-WTA Ensemble on signals with different SNR values.

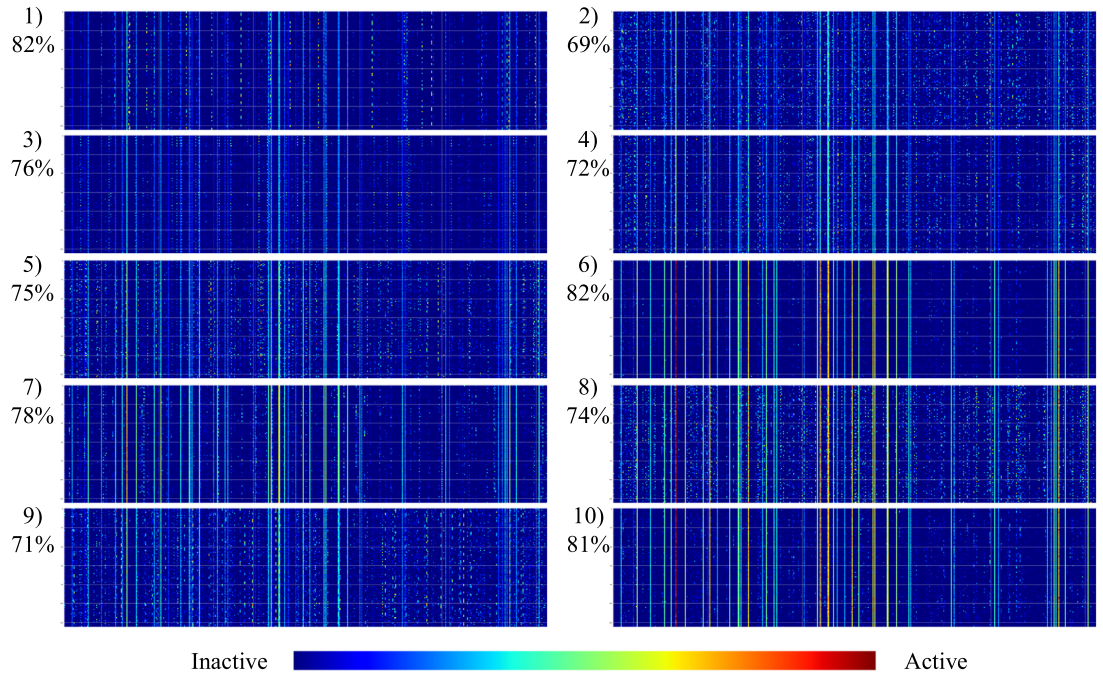


FIGURE 12. Activations of hidden neurons of 1250 samples under load 1, number 1) to 10) represents ten health categories of the bearing, and the percentage denotes the average sparsity of the batch.

model by visualizing the weights and activations of the encoder.

In Fig. 12, the activations of hidden neurons of 1250 samples under load 1 are plotted. For each category there are 125 samples, and number 1) to 10) represent normal, three levels of ball fault, three levels of inner race fault and three levels of out race fault, respectively. In addition, the percentage beside the plot is the average sparsity (percentage of zero activations in this batch of samples). The average sparsity values are all bigger than 69%, which shows that FC-WTA are learning really sparse features of the inputs. From the plots of activations of hidden neurons, we can see

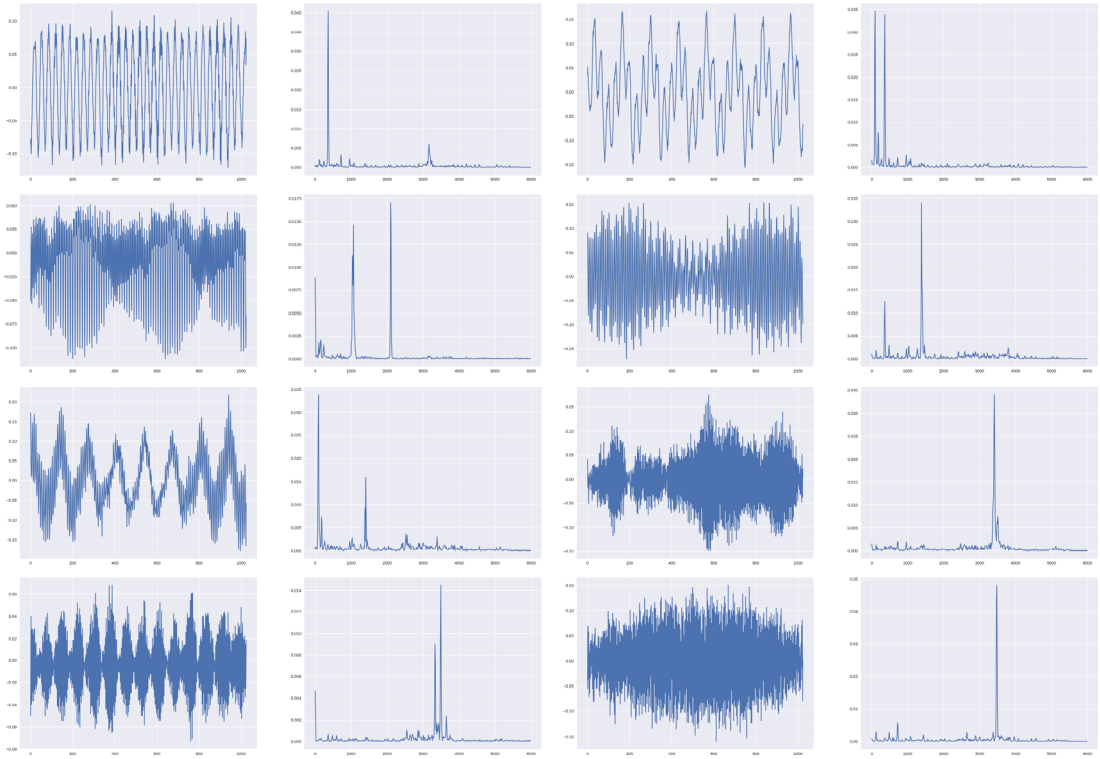


FIGURE 13. Examples of weights and frequency representations of neurons that correspond to sparse features.

some differences between categories. Healthy samples have the highest average sparsity, and from the plot we also see that it has the least number of neurons activated compared to fault signals. Compared to healthy signals, the plots of fault signals have many large activations scattering across different neurons. It is also very interesting to see that in all the plots, there are some ''vertical stripes'' occurring all the time, but their colors vary in different categories, which suggest these features are common in all categories, and their values vary in different categories.

According to the average sparsity value of each hidden representations in all 1250 samples of load 1, we can roughly classify the 512 hidden representation into three categories: 1) 150 extremely sparse features including dead neurons (average sparsity bigger than 0.9); 2) 309 sparse features (average sparsity smaller than 0.9 but bigger than 0.5); and 3) 53 dense features including neurons that are constantly activated (average sparsity smaller than 0.5). The dense features can be seen in Fig. 12 which appears like the vertical stripes shown, and 309 sparse features are the scattered activations in the plots. The dense features are activated for almost all categories, so it's hard to used them to distinguish between fault types. The sparse features, however, are only activated for a small amount of samples, which will make different categories much easier to be divided with the classifier. Some examples of the weights and their frequency representations computed by FFT of hidden neurons that correspond to sparse features are shown in Fig. 13.

V. CONCLUSION

This paper proposes a new bearing fault diagnosis model based on fully-connected winner-take-all (FC-WTA) autoencoder. FC-WTA works directly on temporal vibration signals without any time-consuming feature engineering process. Contributions of this paper include the use of winner-take-all during training stage to learn sparse features suitable for bearing fault diagnosis, and soft-voting during testing stage to increase diagnosing accuracy and stability.

The proposed FC-WTA method explicitly place lifetime sparsity constraint on hidden layer, and then use L_1 penalty to further impose population sparsity, and therefore the proposed method learns very sparse features from input signals. The input of the model is also randomly corrupted as a means to increase the model's robustness to noise. In Section IV-B, the influences and choice of these hyper-parameters are explored with experiments.

In addition, an ensemble method is implemented to enhance the accuracy and stability of FC-WTA. We use a sliding window with length being 1024 and stride being 256 to slice a signal of length 2048 into five segments, and then feed them into FC-WTA to get five sofmax outputs. The softmax outputs representing probability of each category are then aggregated be averaging them. This process is called soft voting. In Section IV-C, we show that soft voting can effectively improve the performance of proposed method.

The proposed method is compared with SVM and DNN whose inputs are frequency features extracted by FFT, and

WDCNN whose input is normalized temporal signal on the diagnosis of simulated dataset by adding white Gaussian noise to the original signals. Results shows that, although all these models could achieve pretty high diagnosis accuracy on normal signals or signals with few noise, their performance degrades rapidly under noisy environment. FC-WTA Ensemble, however, can not only achieve high classification accuracy, but also be very robust to noise.

Some simple networks visualization methods are used to investigate the inner mechanism of the proposed FC-WTA model. The network visualization intuitively shows the sparse features extracted by FC-WTA autoencoder by plotting the activations of encoded representations, and the weights and frequency representations of some hidden neurons are shown.

In future work, we can try finding the signals that can maximize the activations of hidden neurons as a means to explore the meaning and functions of the features extracted by FC-WTA autoencoder, so that we may find out why the proposed method can achieve such good robustness to noise with a two-layer structure.

REFERENCES

[1] S. Yin and O. Kaynak, "Big data for modern industry: Challenges and trends [point of view]," *Proc. IEEE*, vol. 103, no. 2, pp. 143–146, Feb. 2015.

[2] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3D object classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 665–673.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[4] L. Deng *et al.*, "Recent advances in deep learning for speech research at Microsoft," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2013, pp. 8604–8608.

[5] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[6] C. Lu, Z.-Y. Wang, W.-L. Qin, and J. Ma, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification," *Signal Process.*, vol. 130, pp. 377–388, Jan. 2017.

[7] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," *Mech. Syst. Signal Process.*, vols. 72–73, pp. 303–315, May 2016.

[8] Z. Huijie, R. Ting, W. Xinqing, Z. You, and F. Husheng, "Fault diagnosis of hydraulic pump based on stacked autoencoders," in *Proc. 12th IEEE Int. Conf. Electron. Meas. Instrum. (ICEMI)*, vol. 1. Jul. 2015, pp. 58–62.

[9] L. Guo, H. Gao, H. Huang, X. He, and S. Li, "Multifeatures fusion and nonlinear dimension reduction for intelligent bearing condition monitoring," *Shock Vibrat.*, vol. 2016, Jan. 2016, Art. no. 4632562.

[10] N. K. Verma, V. K. Gupta, M. Sharma, and R. K. Sevakula, "Intelligent condition based monitoring of rotating machines using sparse auto-encoders," in *Proc. IEEE Conf. Prognostics Health Manage. (PHM)*, Jun. 2013, pp. 1–7.

[11] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors*, vol. 17, no. 2, p. 425, 2017.

[12] M. Amar, I. Gondal, and C. Wilson, "Vibration spectrum imaging: A novel bearing fault classification approach," *IEEE Trans. Ind. Electron.*, vol. 62, no. 1, pp. 494–502, Jan. 2015.

[13] C. Combastel, "Merging Kalman filtering and zonotopic state bounding for robust fault detection under noisy environment," *IFAC-PapersOnLine*, vol. 48, no. 21, pp. 289–295, 2015.

[14] A. Makhzani and B. J. Frey, "Winner-take-all autoencoders," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2791–2799.

[15] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, Feb. 2010.

[16] S. Ioffe and C. Szegedy. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." [Online]. Available: https://arxiv.org/abs/1502.03167

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 2017.

[19] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. ACM 25th Int. Conf. Mach. Learn.*, Jul. 2008, pp. 1096–1103.

[20] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Aistats*, Apr. 2011, vol. 15. no. 106, pp. 315–323.

[21] X. Lou and K. A. Loparo, "Bearing fault diagnosis based on wavelet transform and fuzzy inference," *Mech. Syst. Signal Process.*, vol. 18, no. 5, pp. 1077–1095, 2004.

CHUANHAO LI received the B.A. degree in english and the B.S. degree in mechanical engineering from the Harbin Institute of Technology, Harbin, China, in 2016, where he is currently pursuing the M.S. degree in mechatronics. His research interests include the applications of deep learning in prognostics and health management of mechanical systems, especially bearings.

WEI ZHANG received the B.A. degree in mechanical engineering from the Harbin Institute of Technology, Harbin, China, in 2015, where he is currently pursuing the M.S. degree in mechatronics. His research interests include the applications of deep learning in fault diagnosis, signal processing, and computer vision.

GAOLIANG PENG received the B.S., M.S., and phPh.D. degrees in mechanical engineering from the Harbin Institute of Technology in 2001, 2003, and 2007, respectively. He worked asheld a postdoctoral position at the Department of Computer Science, Harbin Institute of Technology., where He he is currently an Associate Professor and the Director of the Faculty of Mechatronics and Automation. His research interests include CAD/CAM, robotic servo control, and automatic assembly of mobile radar antenna.

SHAOHUI LIU (M'08) received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology (HIT), Harbin, China, in 2000, 2002, and 2007, respectively. He is currently an Associate Professor with the Department of Computer Science, HIT. His research interests include computer vision, pattern recognition, and image and video processing.