Springer

KSME

Journal of Mechanical Science and Technology 33 (11) (2019) 5177~5188
www.springerlink.com/content/1738-494x(Print)/1976-3824(Online)
DOI 10.1007/s12206-019-1007-5

A new fault diagnosis method based on convolutional neural network and compressive sensing †

Yunfei Ma, Xisheng Jia*, Huajun Bai, Guozeng Liu, Guanglong Wang, Chiming Guo and Shuangchuan Wang

Shijiazhuang Campus, Army Engineering University, Shijiazhuang 050003, China

(Manuscript Received May 29, 2019; Revised September 1, 2019; Accepted September 19, 2019)

Abstract

Compressive sensing is an efficient machinery monitoring framework, which just needs to sample and store a small amount of observed signal. However, traditional reconstruction and fault detection methods cost great time and the accuracy is not satisfied. For this problem, a 1D convolutional neural network (CNN) is adopted here for fault diagnosis using the compressed signal. CNN replaces the reconstruction and fault detection processes and greatly improves the performance. Since the main information has been reserved in the compressed signal, the CNN is able to extract features from it automatically. The experiments on compressed gearbox signal demonstrated that CNN not only achieves better accuracy but also costs less time. The influencing factors of CNN have been discussed, and we compared the CNN with other classifiers. Moreover, the CNN model was also tested on bearing dataset from Case Western Reserve University. The proposed model achieves more than 90 % accuracy even for 50 % compressed signal.

Keywords: Compressive sensing; Fault diagnosis; Convolutional neural network; Feature extraction; Gearbox; Bearing

1. Introduction

With the advancement of information processing technology, traditional repair and preventive maintenance are gradually eliminated and replaced by condition-based predictive maintenance (CBPdm). An efficiency monitoring system is required to guarantee the core components operating normally. In such a system, high-speed data transmission is the key to intelligent management and maintenance. Facing fast sampling rates and various monitoring requirements, there are great needs for methods that can reduce the burden of data saving and transmitting.

Several algorithms have been proposed for signal compression, such as wavelet transform [1], arithmetic coding [2], and Huffman coding [3]. Nevertheless, compressive sensing (CS) [4-6] theory provides a new idea for solving this problem. CS framework first represents the original signal by a low-dimension signal which preserves the main information. Then, the original signal can be acquired by appropriate methods, such as convex optimization algorithms [7], greedy-based algorithms [8]. In a CS monitoring system, the vibration signal is acquired and compressed in a sensor node, and then transmitted to the upper node. The signal reconstruction will be conducted in the upper node for approximately getting the original signal.

During the past few years, the application of CS in machinery monitoring and fault detection has emerged. Sun et al. [9] adopted the block sparse Bayesian learning method for bearing data compression. They also illustrated the efficiency of this algorithm by computing the fault classification accuracy. Zhang et al. [10] applied CS and K-SVD theory to fault detection. Special sparse dictionaries for each fault were trained and then testing data were classified according to the results of decomposition. In referring Ref. [11], a sampling strategy was proposed for preserving the harmonics information. As harmonics often represent the fault frequencies in a vibration signal, the fault features can be derived from the compressed data before reconstruction. Wang et al. [12] proposed a parallel FISTA-like proximal decomposition algorithm under the CS framework. The experiments show that their method can retain time-frequency signatures using only small measurements.

Although the CS reduces data size during transmission, it costs great time for signal reconstruction in the host computer. Since it is possible to reconstruct the original signal through the information preserved in the low-dimension signal, we can consider achieving fault classification using the compressed signal only. This is the starting point of our paper.

*Corresponding author. Tel.: +86 31187994538
E-mail address: xs_jia@hotmail.com
† Recommended by Associate Editor Doo Ho Lee

Improved by Hinton et al., deep learning [13] has become a hotspot in machine learning and artificial intelligence. Deep learning has strong capabilities in adaptive feature learning due to its multi-layer network structure. Convolutional neural network (CNN), as a new deep learning technique, has been developed and widely applied in fault diagnosis of machinery recently. Traditional fault diagnosis algorithm consists of three parts: feature extraction, feature selection and fault classification. The feature extraction methods include time-frequency analysis, cepstrum analysis, wavelet transform, empirical mode decomposition, etc. Those methods highly rely on personal judgments. However, CNN can realize automatic feature extraction and deep learning through hidden layers. In recent years, new CNN architecture, such as AlexNet [14], GoogleNet [15], ResNet [16], continues to win the championship of the image recognition contest ImageNet.

Traditional CNN are mainly used for the recognition of two-dimensional images. However, researchers have proposed a 1D CNN that can be used for vibration signal detection. Ince et al. [17] fused the feature extraction and fault classification into a single body using 1D CNN. The proposed method eliminates the need of complex feature extraction algorithms. Zhou et al. [18] adopted the modified 1D convolutional kernel and pool layers to adapt 1D domain signals. Wu et al. [19] tested the 1D CNN model with the compound fault data from a tank gearbox. The results show that the precision of proposed model is higher than traditional fault diagnosis methods. Wen et al. [20] converted the 1D signal to 2-D images and tested the method using three famous datasets. The advantage of CNN is that it eliminates the effect of handcrafted features. Zhu et al. [21] utilized raw data from sensors and put them into CNN model. Aiming at the typical time shift property of vibration signal, they proposed the strategies of weight summing and large-scale maximum value pooling. It has been demonstrated that the shift invariant CNN has a higher accuracy than traditional fault diagnosis methods.

For reducing the time consumption of host computer, this paper proposed a CNN-based fault detection framework using the CS compressed signal. The 1D CNN was directly applied to the CS compressed signal without reconstruction. According to CS theory, main information of original signal was preserved in the compressed data. Thus, 1D CNN can extract the fault features in CS compressed signal automatically. This method is tested on vibration signal from planetary gearbox test rig and bearing dataset from Case Western Reserve University. Some interesting conclusions have been obtained during experiments.

The main contributions of this paper are outlined as follows: (1) This paper is the first application of CNN model on compressed signal directly. The 1D CNN framework is proposed for replacing traditional reconstruction and fault detection processes, and it has been demonstrated that the CNN has the ability of feature extraction from the compressed signal. (2) The effectiveness and influencing factors of CNN are discussed and CNN is compared with other classifiers. The structure of this paper is as follows. In Sec. 2, CS and CNN theory is briefly recalled. The new framework of 1D CNN on compressed signal is then given in Sec. 3. Sec. 4 presents the experiments of proposed method using compressed gearbox and bearing signal. Sec. 5 summarizes the paper.

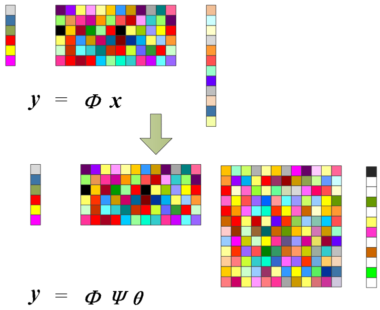


Fig. 1. The framework of compressive sensing.

2. Related work

2.1 Compressive sensing

The Nyquist sampling principle tells us the sampling frequency f_s have following relationship with the signal frequency f to guarantee the quality of sampling.

$$f_s \geq 2f \ . \tag{1}$$

However, real-time monitoring with high sampling places a great burden on the sensors and transmission. To solve this problem, compressive sensing [4] uses a low-dimension signal to approximate the original signal. This can be seen as a breakthrough of Nyquist sampling. Because the vibration signal is not sparse, we need to decompose it with a sparse basis before compression. Assuming $\boldsymbol{\Psi} \in R^{N \times N}$ as the sparse transform, original signal $\boldsymbol{x}$ ($\boldsymbol{x} \in R^N$) can be represented as

$$\boldsymbol{x} = \boldsymbol{\Psi\theta} \ , \tag{2}$$

where $\boldsymbol{\theta}$ ($\boldsymbol{\theta} \in R^N$) is the coefficients vector in $\boldsymbol{\Psi}$-domain. Assuming the length of measurement is M, the compression ratio (CR) can be defined as:

$$CR = \frac{N-M}{N} \ . \tag{3}$$

If $\boldsymbol{x_0}$ represents N-M smallest coefficients of $\boldsymbol{x}$ set 0, then we have $\|\boldsymbol{x_0}\text{-}\boldsymbol{x}\|_2 / \|\boldsymbol{x}\|_2$ negligibly small when $M << N$. Based on this observation, the CS measurements may be represented as

$$\boldsymbol{y} = \boldsymbol{\Phi x} = \boldsymbol{\Phi\Psi\theta} = \boldsymbol{\Theta\theta} \ , \tag{4}$$

where $\boldsymbol{y}$ ($\boldsymbol{y} \in R^M$) represents compressed measurement, $\boldsymbol{\Phi}$ ($\boldsymbol{\Phi} \in R^{M \times N}$) represents the measurement matrix, and $\boldsymbol{\Theta}$ ($\boldsymbol{\Theta} \in R^{M \times N}$) is called the sensor matrix in CS. Solving a sparse vector $\boldsymbol{\theta}$ is a commonly discussed problem [6, 7] with respect to $\boldsymbol{\Theta}$. The

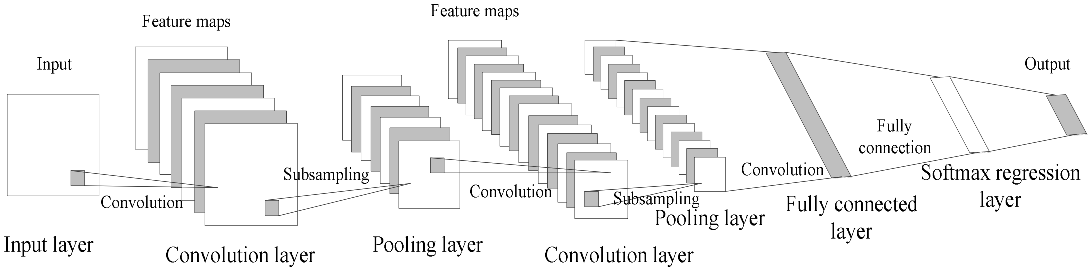


Fig. 2. Structure of classical LeNet-5 CNN.

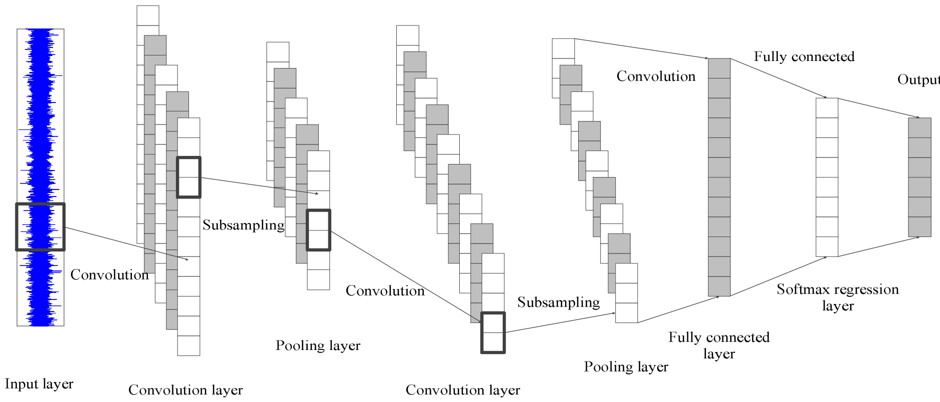


Fig. 3. Structure of 1D CNN.

framework of CS is shown in Fig. 1. Mathematically, this is a 0-norm optimization problem that is NP-hard. The solving methods include orthogonal matching pursuit (OMP) [8] and basic pursuit (BP) [22].

Accurate reconstruction of sparse signal requires restricted isometric property (RIP) of measurement matrix $\boldsymbol{\Phi}$, that is

$$(1-\delta_k)\|\boldsymbol{s}\|_2^2 \le \|\boldsymbol{\Phi s}\|_2^2 \le (1+\delta_k)\|\boldsymbol{s}\|_2^2 , \tag{5}$$

where $\boldsymbol{s}$ represents a random k-sparse vector, and $\delta_k \in (0,1)$. The reconstruction problem is then converted to a 1-norm optimization problem when measurement $\boldsymbol{\Phi}$ satisfies Eq. (5). The convex optimization problem can be solved by Lasso [23], LARS et al. [24].

2.2 Convolutional neural network

Shallow machine learning algorithms use manually obtained features for classification or prediction because they cannot extract features from the complex data. However, feature extraction heavily depends on people's experience. Once the feature is not suitable or the number is insufficient, machine learning will be difficult to achieve the desired results. Compared with shallow learning, deep learning can get the expression of complex functions by training a deep nonlinear network structure which has better generalization ability. Deep learning includes three models: Deep belief network (DBN), convolutional neural network (CNN), and deep neural network (DNN). CNN is a feedforward neural network inspired by mammalian visual multi-layer cells. Different from the fully connected artificial network, the neuron in each layer is only sparsely connected to a certain set of neurons from the previous layer. The theory is widely used in the field of image recognition and has almost monopolized the ImageNet competition. The most famous CNN models include LeNet-5, AlexNet, and GoogleNet. The classical LeNet-5 [25] CNN network consists of 1) an input layer, 2) alternating convolution and 3) pooling layers, and finally 4) a fully connected layer, 5) a softmax regression layer, and 6) an output layer. Fig. 2 shows this CNN network. While the AlexNet, proposed by Alex Krizhevsky, greatly increases the depth of network. Moreover, it adopts the Relu function and local response norm (LRN) layer for the first time.

There are several feature maps in a convolutional layer, and each feature map is obtained by convolving the data of previous layer by a corresponding convolutional kernel. The process of convolution can be represented as

$$g(i)=\sum_{x=1}^{m}\sum_{y=1}^{n}\sum_{z=1}^{p}c_{x,y,z}\times w_{x,y,z}^{i}+d^{i} , \tag{6}$$

where $g(i)$ is the feature map of i-th convolutional kernel, c represents the input data of previous layer and d represents the bias of the kernel. In Eq. (6), x, y, z represent the dimension of input data. After the convolution is completed, a nonlinear transformation is performed with activation function.

The pooling layer implements two tasks: One is to reduce

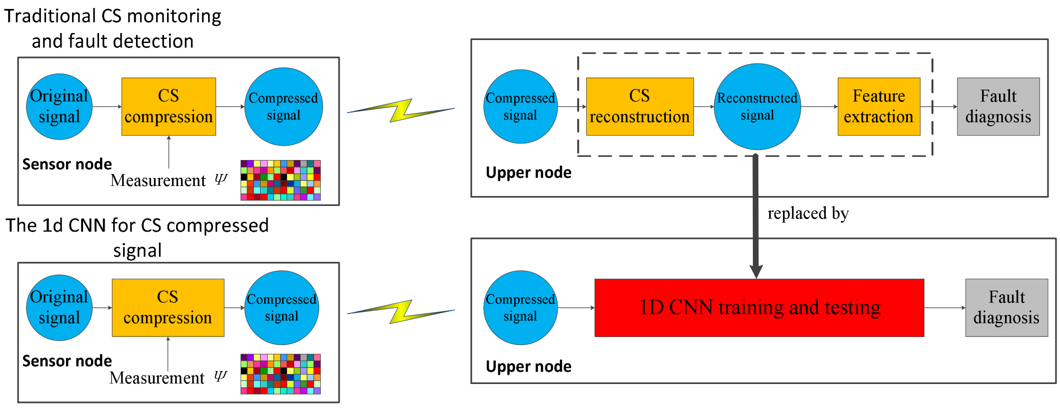


Fig. 4. Comparison of 1D CNN for CS compressed signal with traditional monitoring and fault detection.

the dimension of the feature graph and save calculation, and the other is to control overfitting. Typical pooling layers include maximum pooling, average pooling and 2-norm pooling. After convolution and pooling, a fully connected layer is performed and this layer can use different classification models.

With the depth increasing, the computational work of convolutional neural network increases exponentially. Therefore, in mechanical fault diagnosis, a five-layer convolutional neural network is commonly used for implementing feature learning and classification.

3. 1D CNN for CS compressed vibration signal

This section introduces a feature learning and fault diagnosis model based on 1D CNN for compressed vibration signal. The model performs adaptive feature learning through multiple alternating convolutional and pooling layers, and combines the fully connected layer to achieve fault diagnosis. In this paper, a classical LeNet-5 model has been improved to achieve 1D CNN. The model consists of two convolution layers, two pooling layers and one fully connected layer. The structure of 1D CNN is shown as Fig. 3. As for CNN structure, the following experiments have demonstrated that the LeNet-5 model is efficient enough to achieve 97.5 % success rate, and there is no need for adding layers. Compared to 2D CNN, the 1D CNN model has the following improvements:

(1) The 1D convolution kernel and pooling kernel have been used for 1D vibration signal. For 1D time domain signals, 1D convolution kernel and pooling kernel can directly process the original signal without pretreatment.

(2) The first convolution layer uses large convolution kernel to extract time domain signal features because the signal length is big. However, the size of kernel decreases as depth increasing.

Then we try to use the 1D CNN to process the CS compressed signal. The CS algorithm uses a preset observation matrix to perform dimensional reduction projection on the original signal to achieve data compression. The compression and reconstruction of vibration signal will cost great complexity and running time. Therefore, we consider combining the

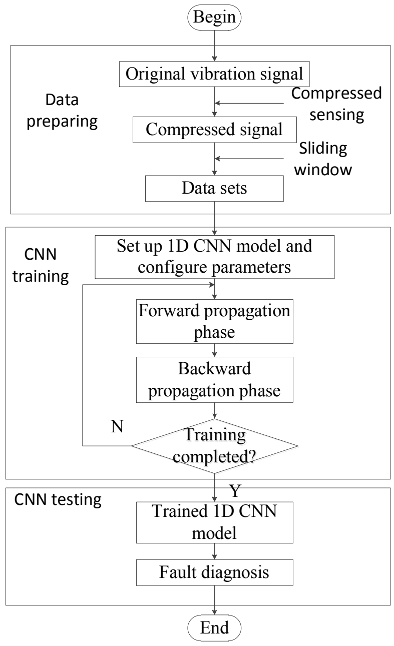


Fig. 5. The flowchart of the proposed method.

data compression and fault diagnosis. In this way, the fault diagnosis can be achieved directly without signal reconstruction. Fig. 4 compares the new monitoring and fault detection framework with the traditional one.

The flowchart of the proposed method can be summarized as Fig. 5. Since the main features of the original signal are preserved in the compressed signal, CNN can easily extract the features and complete fault classification using compressed signal. Considering that CNN training requires a large number of samples, the sliding window method is used here for compressed data sampling. Thus, there is a certain overlap between adjacent samples. The whole algorithm works as follows:

(1) Data preparing. First, the original signal is split into signal blocks using a sliding window. Then, each signal block is compressed with a preset measurement $\boldsymbol{\Phi}$. The details of this part will be described in Sec. 4.

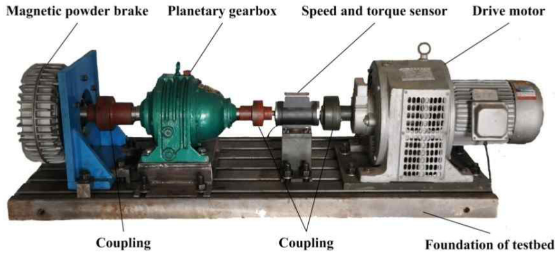


Fig. 6. Planetary gearbox test rig.

(2) CNN training. We set up a CNN model and configure parameters, including learning rate, epochs (training number), batch size, convolution kernel size. Then, we use the training samples made in the last step to train this CNN model. The training process consists of two steps: Forward propagation and Backward propagation. Through error back propagation, the CNN model is optimized by the gradient descent method. The residual is defined as the effects of the parameters on the error of the final output. It is calculated after each training, and the related parameters are updated to realize the automatic learning. The iterative processes are repeated until training finishes.

(3) CNN testing. After the CNN model is trained, we use this model to test the unknown samples and calculate the classification accuracy.

4. Experiment

4.1 Case 1: Planetary gearbox fault diagnosis

The proposed fault diagnosis method is validated using signals collected form the planetary gearbox. Fig. 6 shows the planetary gearbox test rig. The sampling frequency of the test rig is 20 KHz and the experiments are conducted under the following speeds: 400 rpm, 800 rpm and 1200 rpm. For each speed, four kinds of loads are applied, 0 Nm, 0.4 Nm, and 0.8 Nm.

The length of training sample is set to 4000 because the CNN requires a complete cycle for diagnosis. Each type of the training samples contains 300 signals which can be divided into three groups, and each group contains 100 signals according to the different speeds and loads. All types of training samples and testing samples are shown in Table 1.

We used compressed signals rather than the raw data for failure diagnosis. The preset failures are introduced to the gears and they are seeded on one tooth of sun gear, planet gear and ring gear, respectively. The specific failures are shown in Fig. 7. According to the failure position, the gearbox signals can be divided into four categories: 1. Normal state, 2. Planet gear failure, 3. Ring gear failure and 4. Sun gear failure.

The preparation of training samples and testing samples consists of two steps: (1) Signal split using sliding window and (2) signal compression. For sufficient data, the original signal is swept by a sliding window, and the step size is smaller than the window size. Thus, a part of data between adjacent windows is duplicated, which increases the data amount, as shown in Fig. 8. According to Sec. 2, the design of measurement matrix $\boldsymbol{\Phi}$ must follow the RIP criterion. The most commonly used measurement matrix includes the Gaussian random matrix, Bernoulli matrix, and generalized orthogonal combination matrix. We selected the Gaussian random matrix as the measurement matrix in this paper. Candes [25] proves that matrix $\boldsymbol{\Phi}$ satisfies the RIP criterion with a large probability when it is a random Gaussian matrix.

The compression ratio CR is set to 25 %; thus the compressed signal block length is 3000. The parameters of CNN

Table 1. Training and testing samples.

State of signal	Load (Nm)	Speed (rpm)	Number of training samples	Number of testing samples
Normal state	0	400	100	20
	0.4	800	100	20
	0.8	1200	100	20
Planet gear failure	0	400	100	20
	0.4	800	100	20
	0.8	1200	100	20
Ring gear failure	0	400	100	20
	0.4	800	100	20
	0.8	1200	100	20
Sun gear failure	0	400	100	20
	0.4	800	100	20
	0.8	1200	100	20

Table 2. The configurations of CNN model.

Parameters	Value
Input	3000×1
C1: Feature maps	8@2894×1
S2: Feature maps	8@1447×1
C3: Feature maps	8@1395×1
S4: Feature maps	8@279×1
Learning rate	0.2
Batch size	4
Epochs	100


(a) (b)


(c)

Fig. 7. Seeded wear failure: (a) Sun gear; (b) planet gear; (c) ring gear.

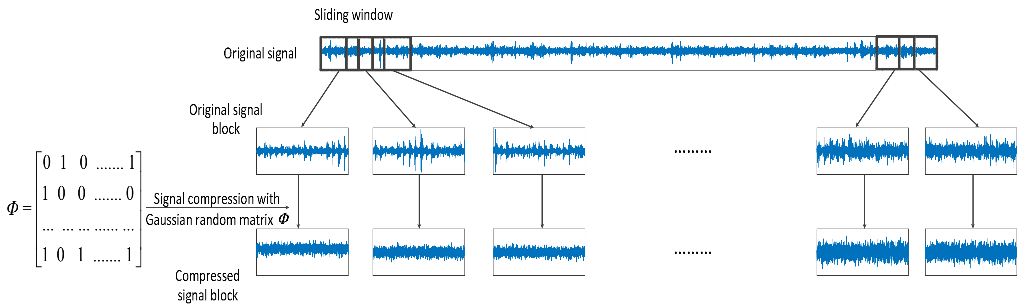


Fig. 8. The process of signal preparation.

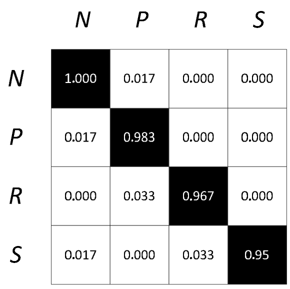


Fig. 9. The confusion matrix of CNN on planetary gearbox data.

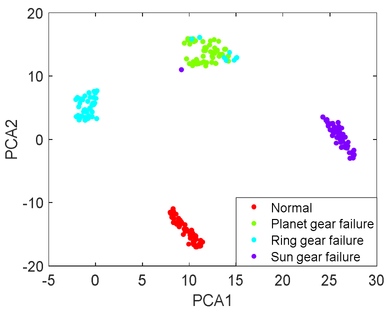
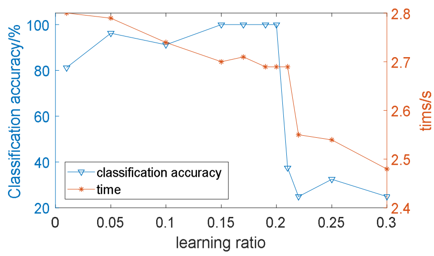
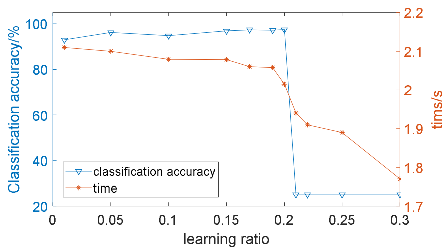


Fig. 10. PCA analysis of the 1D CNN learned features.



(a)



(b)

Fig. 11. The CNN performance with learning ratio increasing for (a) original signal; (b) compressed signal.

are listed in the Table 2. C1, C2 represent the first and second convolution layer while S2, S4 represent the first and second maximum pooling layer of CNN. The results show that CNN based on compressed signal has 97.5 % average accuracy, and the average running time is 188.93s. The confusion matrix of the CNN classification is shown as Fig. 9, and the four states are represented as '*N*', '*P*', '*R*', '*S*', respectively. As shown in Fig. 9, the latter states are easily to be confused with the previous states, which may be caused by the learning order of CNN. The confusion matrix of case 2 also demonstrated this conclusion.

Moreover, we tried to visualize the feature learning ability of the 1D CNN. A principal component analysis (PCA) was used to reduce the dimension of feature matrix obtained from the fully connected layer and the dimension was reduced to 2. As shown in Fig. 10, the four states are highly separated considering two principal components (PC), which indicates good classification performance of 1D CNN. Therefore, we can obtain the following conclusion: 1D CNN can extract features from compressed signal and achieve states classification.

4.1.1 Effect of learning rate

Normally, a lower learning rate means more learning time and a better performance of CNN. The experiments based on original signal and compressed signal have demonstrated this assumption. The effects of learning rate on classification and running time are shown in Fig. 11. From the results, we find that the performance dropped sharply at around 0.2. This means that the learning rate of 0.2 is a critical value.

From Fig. 11, we find that the CNN classification achieves better accuracy when learning ratio is small. But the compression ratio should not be too small, otherwise the success rate will decrease.

4.1.2 Comparison of 1D CNN and 2D CNN

In this section, we compared the time and precision of 1D CNN and 2D CNN and show the outcomes. To verify the efficiency of the comparison, we set the same amounts of layers (2), batch size (5), and training number (30).

The number of training and testing samples of the two mod-

Table 3. Comparison of 1D CNN and 2D CNN using the same planetary gearbox sample.

Compression ratio	1D CNN		2D CNN	
	Time/s	Accuracy	Time/s	Accuracy
10 %	269	96.2 %	146	80.9 %
20 %	283	95.4 %	151	73.1 %
25 %	277	92.5 %	154	72.5 %
30 %	216	90.3 %	112	67.9 %
40 %	217	72.6 %	115	33.1 %

Table 4. The relationship between convolution kernel size and precision of fault classification (learning ratio = 0.2, batch size = 5, CR = 25 %, L = 3000).

Convolution kernel size	87-43	93-45	99-47	101-51	107-53	113-55
Fault classification accuracy	25 %	25 %	25 %	65 %	97.5 %	95 %
Convolution kernel size	119-57	120-60	125-59	127-63	131-61	133-65
Fault classification accuracy	82.5 %	34 %	25 %	25 %	25 %	25 %

Table 5. The relationship between convolution kernel size and precision of fault classification (learning ratio = 0.2, batch size = 5, CR = 25 %, L = 4000).

Convolution kernel size	125-59	131-61	133-65	139-67	145-69	151-71
Fault classification accuracy	30 %	40 %	25 %	25 %	35 %	25 %
Convolution kernel size	153-75	159-77	165-79	171-81	177-83	183-85
Fault classification accuracy	50 %	92.5 %	97.5 %	55 %	25 %	25 %

Table 6. The relationship between convolution kernel size and precision of fault classification (learning ratio = 0.2, batch size = 5, CR = 25 %, L = 5000).

Convolution kernel size	151-71	153-75	159-77	171-81	177-83	183-85
Fault classification accuracy	25 %	25 %	25 %	40 %	20 %	45 %
Convolution kernel size	185-89	189-87	191-91	197-93	201-95	205-97
Fault classification accuracy	95 %	97.5 %	40 %	25 %	25 %	25 %

els is completely the same, so the length of the sample is the same. We converted the 1D 784 length signal into 2D 28×28 images. From Table 3, we can see that the running of 2D CNN is less time consuming compared to 1D CNN. However, the accuracy of 1D CNN is higher than 2D CNN within the same circumstance. This may be due to the vibration signal structure that is destroyed in the process of converting to 2D CNN, thereby resulting in the loss of some feature information. Since the vibration signal is a 1D signal, we prefer a 1D CNN to maintain the signal structure.

4.1.3 Effect of kernel size and amount

In the past experiments we set the kernel size of two layers as 107-53, which achieved good results. The number of convolutional kernels for each layer should be an integer multiple of 8. And we set the number of convolutional kernels as 8, 8. If the length of convolution kernel is too small, the recognition rate is low. While if it is too large, an overfitting will occur. Thus, the choice of kernel size decides the efficiency of CNN directly. Normally, the kernel size is dependent on the length of each data sample.

We tested the relationship between the best convolution kernel size and the length of data sample as follows. The learning ratio, batch size, and compression ratio were set to fixed values. We changed the data sample length L from 3000, 4000 to 5000. From Tables 4-6, we find that the best convolution kernel sizes for three data sample lengths are 107-53, 165-79, and 189-87. The best kernel size is increasing with the sample length growing. Therefore, the convolution kernel size has a great relationship with the data sample length.

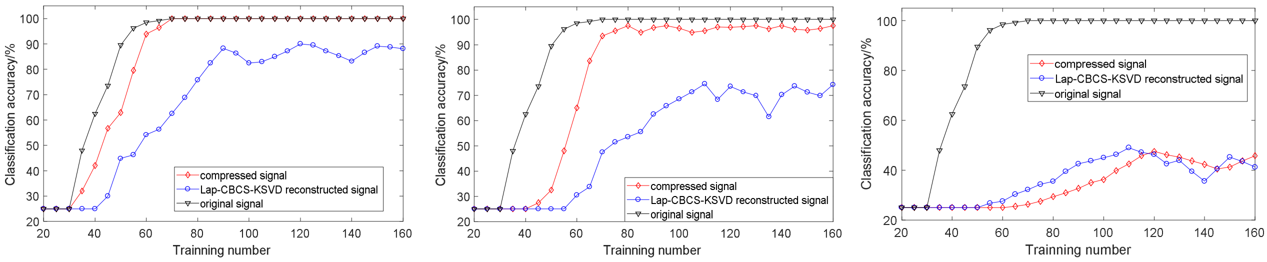
4.1.4 Comparison with CNN on reconstructed signal

In previous study, CNN was used for fault diagnosis based on the compressed signal. In this section, we compared the effects of CNN based on both compressed and reconstructed signal.

The CS reconstruction methods can be classified into three categories: Convex optimization algorithms [7], greedy-based algorithms [8], and Bayesian algorithms [26-28]. Among them, the Bayesian algorithm achieves the best performance. Bayesian compressive sensing (BCS) uses Bayesian estimation to obtain the maximum posterior probability of the original signal.

Table 7. The comparison of time and accuracy for CNN fault diagnosis on compressed signal and reconstructed signal.

Compression ratio	Reconstructed signal					Compressed signal		Improvement	
	Time/s				Fault classification accuracy/%	Time/s	Fault classification accuracy/%	ΔTime/s	Δaccuracy/%
	K-SVD	Lap-CBCS reconstruction	CNN	Total		CNN			
10 %	34	2874	219	3127	87.1	199	100	2928	12.9
15 %	32	2186	214	2432	85.3	195	99.1	2237	13.8
20 %	29	1348	210	1587	82.5	193	98.8	1394	16.3
25 %	34	1035	207	1276	75.3	189	97.5	1087	22.2
30 %	31	982	206	1219	68.2	176	92.9	1043	24.7
35 %	32	947	203	1182	62.0	165	82.5	1017	20.5
40 %	30	928	201	1159	58.5	150	75.7	1009	17.2
45 %	34	762	190	986	52.3	143	59.9	843	7.6
50 %	30	541	189	760	49	140	47.5	620	-1.5
55 %	28	479	184	691	40.8	133	35.9	558	-4.9
60 %	33	457	178	668	25	121	25	547	0



(a) (b) (c)

Fig. 12. Comparison of compressed signal and Lap-CBCS-KSVD reconstructed signal using (a) 10 % compressed signal; (b) 25 % compressed signal; (c) 50 % compressed signal.

In BCS theory, a Gaussian prior-based model is widely used. However, the recent results of Ref. [29] show that Laplace prior-based model may have better sparseness promotion. In this paper, we introduce the Laplace prior-based model to the multitask BCS [27], and call it Lap-CBCS method. For better sparse representation, we changed the traditional sparse transform bases with an over-complete dictionary. The basic idea is to use the K-SVD algorithm [30] for training various signal samples and adaptively update the dictionary.

In summary, we used a Lap-CBCS-KSVD method for signal reconstruction. The algorithm can be regarded as one of the most accurate reconstruction algorithms so far. And after that, we used the reconstructed signal samples for CNN training and testing. We selected three kinds of data for comparison: 10 % compressed signal (*CR* = 10 %), 25 % compressed signal (*CR* = 25 %), and the 50 % compressed signal (*CR* = 50 %). Fig. 12 reveals following phenomenon:

(1) The results of CNN fault diagnosis based on compressed signal are better than that based on reconstructed signal when *CR* = 10 %, 25 %. Although the Lap-CBCS-KSVD is an efficient reconstruction algorithm, the reconstructed signal still has many differences with the original signal. The CNN method may be sensitive to those errors and cause the bad effects of fault diagnosis. Therefore, we reach an interesting conclusion that the CNN using compressed signal may achieve better performance than using reconstructed signal when the compression ratio is relatively small.

(2) According to Fig. 12(c), there are some places where the classification accuracy of CNN based on compressed signal is the lowest. At this time, the compressed signal loses too much information of the original signal and the reconstructed signal can restore the original signal with higher precision.

Table 7 lists the time and accuracy for CNN fault diagnosis in order to compare the compressed signal and reconstructed signal. We conducted 100 experiments for each compression ratio, and calculated the average value for running time and fault classification accuracy. From the results, it can be seen that the CNN with compressed signal is inferior to CNN with reconstructed signal when compression ratio is less than 45 %. The CNN using compressed signal not only has a better fault classification accuracy, but also costs less time. According to Table 7, the time consumption of the compressed signal is far less than the reconstructed signal.

However, when the compression ratio is increasing, the reconstructed signal comes better than the compressed, signal

Table 8. Comparison results of CNN with other methods.

Methods	CR = 10 %		CR = 25 %		CR = 40 %		CR = 50 %		CR = 60 %	
	Mean accuracy/%	Running time/s	Mean accuracy/%	Running time/s	Mean accuracy/%	Running time/s	Mean accuracy/%	Running time/s	Mean accuracy/%	Running time/s
CNN	99.37±0.11	199	97.5±1.76	189	75.7±3.82	150	47.5±4.21	137	25	121
DBN	95.17±0.26	265	92.5±3.16	235	64.5±4.77	196	37.8±6.39	174	25	167
DNN	97.29±0.43	178	93.45±2.78	162	66.35±4.18	143	40.33±5.53	128	25	111
ANN	83.70±3.75	164	65.8±5.42	138	41.87±7.69	102	25	99	25	93

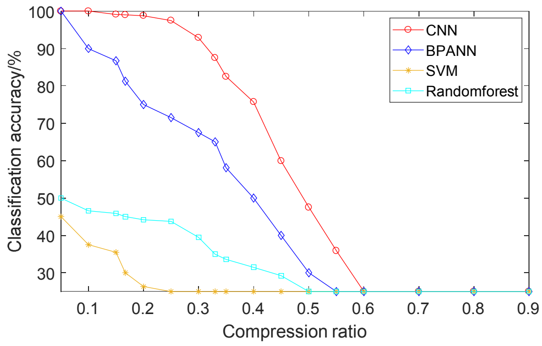


Fig. 13. Comparison of different classification methods.

although it still costs more running time. In this situation, the compressed signal loses too much original information and extracting features from the compressed signal is hard, even using the CNN method. Thus, a relatively good reconstruction will recover some important information, which is a great help for fault classification.

4.1.5 Comparison with other classification methods

For evaluating the performance of proposed deep learning method, other classification methods are selected to test the efficiency in this case. They are BP artificial neural network (BPANN), support vector machine (SVM), and random forest (RF).

The fault diagnosis effects are displayed in Fig. 13, which shows that CNN achieves 100 % fault classification accuracy with the compressed signal, which indicates a feature extraction advantage for CNN. The accuracies of CNN are stable at around 100 % when the compression ratio is less than 0.3. And the BPANN obtains 100 % classification accuracy only when original signal is not compressed at all. However, SVM and random forest are not suitable for fault diagnosis in this case.

Moreover, we compared CNN with other statistical methods and deep learning models. Deep belief network (DBN) [31, 32], artificial neural network (ANN) [33], and deep neural network (DNN) [34, 35] were selected for comparison in this paper. Taking four compressed signals for example, we conducted 100 experiments for each approach and calculated the mean accuracy ρ. And the variance τ was calculated for each method to evaluate the stability. Table 8 gives the range of [ρ-2τ, ρ+2τ] for each method, which can be regarded as the 95 % confidence interval.

Table 9. The 10 kinds of bearing faults selected.

No.	Bearing condition	Fault type	Motor speed (HP)	Motor load
1	Normal	-	0/1/2/3	1797/1772/1750/1730
2	IR	7 mm	0/1/2/3	1797/1772/1750/1730
3	B	7 mm	0/1/2/3	1797/1772/1750/1730
4	OR	7 mm	0/3/1/2	1797/1730/1772/1750
5	IR	14 mm	0/1/2/3	1797/1772/1750/1730
6	B	14 mm	0/1/2/3	1797/1772/1750/1730
7	OR	14 mm	0/1/2/3	1797/1772/1750/1730
8	IR	21 mm	0/1/2/3	1797/1772/1750/1730
9	B	21 mm	0/1/2/3	1797/1772/1750/1730
10	OR	21 mm	1/3/0/2	1772/1730/1797/1750

From Table 8, we can see that with *CR* increasing, the mean accuracy and running time decrease gradually. But the CNN model obtains a good result. The mean accuracy of the 25 % compressed signal is as high as 97.5 %, and it is better than all other methods. Although DNN costs less time, it is slightly less accurate compared to CNN. The DBN costs much more time than others, and it has less accuracy than CNN, DNN. The result of traditional ANN is the worst, which is obviously inferior to the CNN model.

4.2 Case 2: Bearing fault diagnosis

In this section, our proposed 1D-CNN method is performed on the bearing data provided by the Case Western Reserve University (CWRU) [36]. This dataset was produced by deep groove ball bearing from SKF and NTN manufactures. Experiments were carried out under each of the following bearing conditions: Normal, inner race fault (IR), outer race fault (OR), and ball fault (B). The faults were divided into four categories according to the cutting depth: 7 mm, 14 mm, 21 mm, and 28 mm. We selected ten kinds of faults (see Table 9).

There are 800 training samples and 400 testing samples prepared for CNN diagnosis. In other words, each fault type contains 80 training samples and 40 testing samples. To improve the efficiency, the structure was simplified which contains just one alternating convolutional layer and one pooling layer. The configuration of this CNN model is displayed in

Table 10. The configuration of CNN model.

Parameters	Input	C1: Feature maps	S2: Feature maps	Learning rate	Batch size	Number of epochs
Value	2800×1	10@2700×1	10@1350×1	0.2	16	50

Table 11. Training process of 10 % compressed bearing data (CR = 10 %) using CNN model.

Epoch	5	10	15	20	25	30	35	40	45	50
Sample success/%	0.1024	0.3040	0.7119	0.9169	0.9219	0.9317	0.9371	0.9370	0.9419	0.9452
Test success/%	0.1023	0.3040	0.7114	0.9168	0.9217	0.9315	0.9370	0.9367	0.9417	0.9448
Time/s	1.867	1.745	1.740	1.770	1.817	1.751	1.823	1.816	1.753	1.834

Table 12. Training process of 20 % compressed bearing data (CR = 20 %) using CNN model.

Epoch	5	10	15	20	25	30	35	40	45	50
Sample success/%	0.1	0.4110	0.7724	0.9135	0.9135	0.9147	0.9154	0.9152	0.9166	0.9224
Test success/%	0.1	0.4110	0.7724	0.9132	0.9133	0.9144	0.9153	0.9152	0.9164	0.9221
Time/s	1.703	1.758	1.705	1.767	1.788	1.789	1.705	1.770	1.758	1.703

Table 13. Training process of 30 % compressed bearing data (CR = 30 %) using CNN model.

Epoch	5	10	15	20	25	30	35	40	45	50
Sample success/%	0.1	0.3824	0.6525	0.9025	0.9035	0.9044	0.9051	0.9055	0.9071	0.9155
Test success/%	0.1	0.3822	0.6524	0.9024	0.9032	0.9043	0.9049	0.9053	0.9066	0.915
Time/s	0.564	1.667	0.590	0.561	0.558	1.640	0.596	0.578	0.556	0.576

Table 14. Training process of 40 % compressed bearing data (CR = 40 %) using CNN model.

Epoch	5	10	15	20	25	30	35	40	45	50
Sample success/%	0.14	0.2701	0.5634	0.9049	0.9057	0.9057	0.9068	0.9072	0.9079	0.9096
Test success/%	0.14	0.2701	0.5633	0.9046	0.9055	0.9056	0.9065	0.9072	0.9077	0.9094
Time/s	0.508	0.542	0.548	0.541	0.5	0.502	0.545	0.513	0.528	0.525

Table 15. Training process of 50 % compressed bearing data (CR = 50 %) using CNN model.

Epoch	5	10	15	20	25	30	35	40	45	50
Sample success/%	0.12	0.3024	0.5036	0.7155	0.8957	0.8981	0.8994	0.9004	0.9028	0.9032
Test success/%	0.12	0.3023	0.5035	0.7155	0.8957	0.8979	0.8986	0.9001	0.9027	0.9030
Time/s	0.477	0.476	0.443	0.432	0.419	0.448	0.416	0.476	0.475	0.453

Table 16. Training process of 60 % compressed bearing data (CR = 60 %) using CNN model.

Epoch	5	10	15	20	25	30	35	40	45	50
Sample success/%	0.09	0.2416	0.4559	0.6738	0.8882	0.8891	0.8903	0.8916	0.8935	0.8985
Test success/%	0.09	0.2415	0.4558	0.6738	0.8881	0.8889	0.8901	0.8911	0.8934	0.8982
Time/s	0.268	0.346	0.393	0.398	0.405	0.418	0.402	0.426	0.433	0.438

Table 10, taking the 30 % compressed signal for example. The cutting length of original signal is 4000 to insure at least one cycle is included, and the compressed signal is of the length 4000×70 % = 2800. The convolutional kernel size is 101×1 with 10 channels; thus the length of feature map is 2800-101+1 = 2700. The filters of maximum pooling layer are of the size 2×1, and there are ten filters in this layer. The CNN model is written in Python 3.6 with TensorFlow and run on win10 with i5-8250 CPU.

This CNN model was tested using the 10 %, 20 %, 30 %, 40 %, 50 %, 60 % compressed signal. Tables 11-16 display the CNN training processes of those compressed signals. Sample success represents the diagnosis accuracy of training samples, while test success represents the accuracy of testing samples. The average running time for each training was also calculated. The maximum accuracies for those compressed

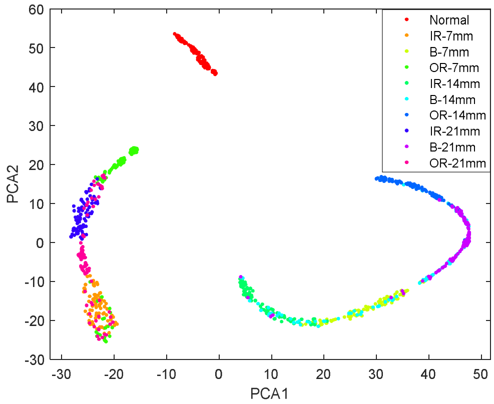


Fig. 14. PCA analysis of bearing data.

	Normal	IR_7mm	B_7mm	OR_7mm	IR_14mm	B_14mm	OR_14mm	IR_21mm	B_21mm	OR_21mm
Normal	0.975	0.000	0.025	0.000	0.000	0.000	0.000	0.000	0.000	0.000
IR_7mm	0.025	0.95	0.025	0.000	0.000	0.000	0.000	0.000	0.000	0.000
B_7mm	0.000	0.05	0.925	0.000	0.025	0.000	0.000	0.000	0.000	0.000
OR_7mm	0.000	0.025	0.05	0.9	0.000	0.025	0.000	0.000	0.000	0.000
IR_14mm	0.000	0.000	0.025	0.000	0.925	0.000	0.025	0.000	0.025	0.000
B_14mm	0.025	0.025	0.000	0.05	0.000	0.875	0.025	0.000	0.000	0.000
OR_14mm	0.000	0.025	0.025	0.000	0.000	0.025	0.9	0.000	0.000	0.025
IR_21mm	0.000	0.000	0.000	0.025	0.000	0.5	0.000	0.925	0.000	0.000
B_21mm	0.000	0.025	0.000	0.025	0.025	0.000	0.025	0.000	0.9	0.000
OR_21mm	0.025	0.025	0.000	0.05	0.000	0.000	0.000	0.025	0.000	0.875

Fig. 15. The confusion matrix of CNN classification on 30 % compressed signal.

signals are 94.48 %, 92.21 %, 91.5 %, 90.94 %, 90.3 %, and 89.82 %.

And can be seen, the diagnosis accuracy grows with the training samples increasing. However, there are few changes when epoch is larger than 25, because no more features can be learned by the CNN model.

Moreover, we found that the training time was decreasing for deeply compressed signal. Fig. 14 shows the PCA analysis of 30 % compressed signal. Fig. 15 shows the confusion matrix of diagnosis results for 30 % compressed signal. The rows of confusion matrix stand for predict labels and columns stand for the actual labels of fault types. It shows that the normal state has 97.5 % accuracy, and B_14 mm, OR_21 mm are the worst two states, which have 87.5 %, 87.5 % accuracy.

5. Conclusion

This paper presents a new fault diagnosis method using the compressed signal. The main contributions of this work are developing a 1D CNN feature extraction and fault detection method, replacing the traditional signal reconstruction and fault diagnosis, and applying this approach to the mechanical fault detection field. According to the experiments on gearbox, our framework obtains a significant improvement over traditional monitoring and fault detection methods on both accuracy and time cost. Moreover, we compared CNN with other classifiers, such as BPANN, SVM, Randomforest, DBN, DNN, and ANN. Experimental results show that CNN is the best for compressed signal processing. The CNN model was also tested on bearing dataset, and it achieved more than 90 % accuracy even for 50 % compressed signal.

The limitations of the CNN include two aspects. First, CNN is a bit slow compared to DNN. The efficiency of CNN needs to be improved in future research. Otherwise, we have found that CNN is sensitive to parameters such as learning rate, batch size, and epoch. A parameters optimization method can be studied for choosing appropriate parameters.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under the grants 71871220. I would like to express my gratitude to all those who helped me during the writing of this paper. A special acknowledgment should be shown to my supervisor Professor Xisheng Jia from whose useful instructions I benefited greatly.

References

[1] M. J. Shi, R. Z. Luo and Y. H. Fu, Fault diagnosis of rotating machinery based on wavelet and energy feature extraction, *J. of Electronic Measurement and Instrumentation*, 29 (8) (2015) 1114-1120.

[2] J. Rissanen and G. G. Langdon, Arithmetic coding, *IBM J. of Research and Development*, 23 (2) (1979) 149-162.

[3] R. Gallager, Variations on a theme by Huffman, *Transaction on Information Theory*, 24 (6) (1978) 668-674.

[4] E. J. Candès, Compressive sampling, *Proc. of International Congress of Mathematics*, Madrid, Spain (2006) 1433-1452.

[5] E. J. Candès and T. Tao, Decoding by linear programming, *IEEE Transactions on Information Theory*, 51 (12) (2005) 4203-4215.

[6] D. L. Donoho, Compressed sensing, *IEEE Transactions on Information Theory*, 52 (4) (2006) 1289-1306.

[7] J. A. Tropp and S. J. Wright, Computational methods for sparse solution of linear inverse problems, *Proceedings of the IEEE*, 98 (6) (2010) 948-958.

[8] J. A. Tropp and A. C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, *IEEE Transactions on Information Theory*, 53 (12) (2007) 4655-4666.

[9] J. D. Sun, Y. Yu and J. T. Wen, Compressed-sensing reconstruction based on block sparse bayesian learning in bearing-condition monitoring, *Sensors*, 17 (2017) 1454.

[10] X. P. Zhang, N. Q. Hu, L. Hu, L. Chen and Z. Cheng, A bearing fault diagnosis method based on the low-dimensional compressed vibration signal, *Advances in Mechanical Engineering*, 7 (7) (2015) 1-12.
[11] G. Tang et al., Compressive sensing of roller bearing faults via harmonic detection from under-sampled vibration signals, *Sensors*, 15 (2015) 25648-25662.
[12] Y. X. Wang et al., Compressed sparse time-frequency feature representation via compressive sensing and its applications in fault diagnosis, *Measurement*, 68 (2015) 70-81.
[13] G. E. Hinton and R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science*, 313 (5786) (2006) 504-507.
[14] A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Proc. of International Conference on Neural Information Processing Systems*, Lake Tahoe, USA (2012) 1097-1105.
[15] C. Szegedy et al., Going deeper with convolutions, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA (2015) 1-9.
[16] K. He et al., Deep residual learning for image recognition, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA (2016) 770-778.
[17] T. Ince et al., Real-time motor fault detection by 1D convolutional neural networks, *IEEE Transactions on Industrial Electronics*, 63 (11) (2016) 7067-7075.
[18] Q. C. Zhou et al., Fault diagnosis for rotating machinery based on 1D depth convolutional neural network, *J. of Vibration and Shock*, 37 (23) (2018) 31-37.
[19] C. Z. Wu, P. C. Jiang and F. Z. Feng, Faults diagnosis method for gearboxes based on a 1-D convolutional neural network, *J. of Vibration and Shock*, 37 (22) (2018) 51-56.
[20] L. Wen et al., A new convolutional neural network based data-driven fault diagnosis method, *IEEE Transactions on Industrial Electronics*, 65 (7) (2017) 5990-5998.
[21] H. J. Zhu et al., Machinery fault diagnosis based on shift invariant CNN, *J. of Vibration and Shock*, 38 (5) (2019) 45-52.
[22] S. Chen, D. L. Donoho and M. A. Saunders, Atomic decomposition by basis pursuit, *SIAM J. on Scientific Computing*, 20 (1) (1999) 33-61.
[23] R. Tibshirani, Regression shrinkage and selection via the lasso, *J. of the Royal Statistical Society*, 58 (1) (1996) 267-288.
[24] M. A. Davenport et al., Signal processing with compressive measurements, *IEEE J. of Selected Topics in Signal Processing*, 4 (2) (2010) 445-460.
[25] E. J. Candes, The restricted isometry property and its implications for compressed sensing, *Comptes Rendus Mathematique*, 346 (9) (2008) 589-592.
[26] S. Ji, Y. Xue and L. Carin, Bayesian compressive sensing, *IEEE Trans. Signal Process*, 56 (6) (2008) 2346-2356.
[27] S. Ji, D. Dunson and L. Carin, Multi-task compressive sensing, *IEEE Trans. Signal Process*, 57 (1) (2009) 92-106.
[28] Z. Zhang and B. D. Rao, Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning, *IEEE J. of Selected Topics in Signal Processing*, 5 (5) (2011) 912-926.
[29] S. Babacan, R. Molina and A. Katsaggelos, Bayesian compressive sensing using Laplace priors, *IEEE Trans. Image Process*, 19 (1) (2010) 53-63.
[30] M. Aharon, M. Elad and A. Bruckstein, K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, *IEEE Transactions on Signal Processing*, 54 (11) (2006) 4311-4322.
[31] H. Shao, H. Jiang, X. Zhang and M. Niu, Rolling bearing fault diagnosis using an optimization deep belief network, *Measurement Science and Technology*, 26 (11) (2015) 115002.
[32] Z. W. Shang, X. X. Liao, R. Geng, M. S. Gao and X. Liu, Fault diagnosis method of rolling bearing based on deep belief network, *J. of Mechanical Science and Technology*, 32 (11) (2018) 5139-5145.
[33] Y. Shatnawi and M. Al-Khassaweneh, Fault diagnosis in internal combustion engines using extension neural network, *IEEE Transactions on Industrial Electronics*, 61 (3) (2014) 1434-1443.
[34] S. Jurgen, Deep learning in neural networks: An overview, *Neural Networks*, 61 (1) (2015) 85-117.
[35] Y. Lecun, Y. Bengio and G. Hinton, Deep learning, *Nature*, 521 (7553) (2015) 436-444.
[36] *http://csegroups.case.edu/bearingdatacenter/pages/download-data-file* (2019).

Yun-fei Ma received his M.Sc. in 2017 from Army Engineering University; now he is a Ph.D. candidate in Army Engineering University. His main research interests include compressive sensing, deep learning, and fault diagnosis.

Xi-sheng Jia received his Ph.D. in 2001 from University of Salford; now he is a Professor and Ph. D supervisor at Army Engineering University. His main research interests include Condition Based Maintenance (CBM), Prognostic and Health Management (PHM).