Hindawi
Shock and Vibration
Volume 2020, Article ID 8819313, 10 pages
https://doi.org/10.1155/2020/8819313

Research Article

An Antinoise Fault Diagnosis Method Based on Multiscale 1DCNN

Jie Cao,[1,2,3] **Zhidong He**,[1] **Jinhua Wang**,[1] **and Ping Yu**[1]

[1]*College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China*
[2]*Engineering Research Center of Urban Railway Transportation of Gansu Province, Lanzhou 730050, China*
[3]*Engineering Research Center of Manufacturing Information of Gansu Province, Lanzhou 730050, China*

Correspondence should be addressed to Jie Cao; caoj@lut.edu.cn

Received 1 September 2020; Revised 18 November 2020; Accepted 25 November 2020; Published 14 December 2020

Academic Editor: Dejan Gjorgjevikj

The bearing state signal collected by the vibration sensor contains a large amount of environmental noise in actual processes, which leads to a reduction in the accuracy of the convolutional network in identifying bearing faults. To solve this problem, a one-dimensional convolutional neural network with a multiscale kernel (MSK-1DCNN) is proposed for the classification information enhancement of the input. A two-layer multiscale convolution structure (MSK) is used at the front of the network. MSK has five convolutional kernels with different sizes, and those kernels are used to extract features with varying resolutions in the original signal. In the multiscale convolution structure, the ELU activation function is used instead of the ReLU function to improve the antinoise ability of MSK-1DCNN, also by adding pepper noise to the training set data to destroy the input data and forcing the network to learn more representative features to improve the robustness of the network. Experimental results illustrate that the improved methods proposed in this paper effectively enhance the diagnostic performance of MSK-1DCNN under intense noise, and the diagnostic accuracy is higher than that of other comparison algorithms.

1. Introduction

Rolling bearings are an essential component and the main factor leading to system failures in rotating machinery. 45%–55% of equipment failures are caused by bearing damage [1]. Every unexpected failure of the bearing may lead to the machine and even the entire system's failure, resulting in huge economic losses and a waste of time. As a major problem of fault diagnosis, rolling bearings' fault diagnosis has attracted researchers' extensive attention. The traditional method of bearing fault diagnosis is to analyze the sensor's vibration signal, then use the intelligent algorithm to extract the fault characteristics of the signal, and finally, use the classification algorithm to detect fault type. With the rapid rising of deep learning and its successful applications in computer vision [2], natural language processing [3], medical image analysis [4], and other fields, intelligent fault diagnosis algorithms based on deep learning have also been rapidly developing in recent years [5, 6]. Deep learning algorithms for bearing fault diagnosis include Autoencoders (AE), Restricted Boltzmann Machines (RBM), and Convolutional Neural Networks (CNN).

Compared with AE and RBM, convolutional neural networks have advantages in processing time series data and vibration signals with variable translation characteristics [7]. Researchers have used one-dimensional convolutional neural networks to directly extract fault features from the original signal to classify faults in recent years. T Ince et al. [8] used a one-dimensional convolution neural network to process the motor's current signal. The proposed one-dimensional convolution network is very effective in the calculation and can be easily and cheaply implemented on hardware systems. Eren [9] used a one-dimensional convolutional neural network to quickly and accurately detect motor bearing faults, with an accuracy rate of 97.1%. Zhang et al. [10] proposed a deep convolutional neural network with a wide first-layer convolution kernel (WDCNN). The proposed method uses a wide convolution kernel in the first convolution layer to extract features form original vibration signal and suppress high-frequency noise and, then, uses small convolution kernels in the next layers of the network to achieve multilayer nonlinearity mapping. AdaBN is used to improve the domain adaptability of the model. In another

paper, Zhang et al. [6] proposed a method called TICNN (Convolution Neural Network with Training Interference) for the problem of a lot of noise and variable operating conditions in the working environment of the bearing. TICNN directly extracts the fault characteristics from the original vibration signal without additional data preprocessing. It has made the following improvements: (1) Convolution kernel dropout is used in the first convolutional layer; (2) small batch training is used in the optimization algorithm, and ensemble learning is used to improve the stability of the network.

The current one-dimensional fault diagnosis model achieves a 100% fault recognition rate under no-noise conditions, showing the powerful feature-extraction ability of convolutional neural networks. However, most of the currently proposed models do not consider the situation that the signal contains noise. The signal collected by sensors in the real working environment contains a lot of noise, which will significantly impact the accuracy of the diagnostic model. Therefore, most models have not achieved good diagnostic accuracy in the presence of noise. To address this problem, we propose a one-dimensional convolutional neural network with multiscale convolution kernels (MSK-1DCNN). MSK-1DCNN directly acts on the original vibration signal, and the feature extraction and fault classification are realized through the convolutional neural network.

The main contributions of the present paper are as follows:

(1) At the front of the network, a single-layer and single-kernel convolution layer is replaced with a two-layer multiscale convolution structure. Through multiple convolution kernels of different scales, MSK-1DCNN can extract discriminative features with varying resolutions from the original signal to obtain better diagnostic results at low SNR than the network using a single-layer single-convolution kernel network.

(2) In the multiscale convolution structure, the ELU activation function is used instead of the ReLU function. The negative part of the ELU activation function is a saturated function, which makes its antinoise ability better. Therefore, using ELU functions can improve the accuracy of the network at low SNR.

(3) Pepper noise is added to the input training data during the network training stage. Pepper noise will increase the complexity of the input signal, so adding pepper noise in the training set can improve the network's feature extraction ability, making the network more robust to noise.

This paper's remainder is organized as follows: in Section 2, an introduction of the CNN is presented. The proposed MSK-1DCNN model is introduced in Section 3. Section 4 presents and discusses the result from different experimental conditions. A comparison is also made with the proposed method. We draw the conclusions in Section 5.

2. Introduction of Convolutional Neural Networks

The convolutional neural network is a multilevel feed-forward neural network, which is usually composed of three types of layers: a convolutional layer, pooling layer, and fully connected layer. The convolutional layer and the pooling layer extract the characteristics of input data through convolution calculation and downsampling operations. Then, the fully connected layer achieves classification or regression task. The fully connected layer has the same structure and calculation method as the traditional feed-forward neural network.

2.1. Convolutional Layer. The convolutional layer learns the features of input data through convolution calculation. It is composed of multiple feature maps. Each neuron of each feature map is connected to a local area of the previous layer of feature maps through a set of weights. This local area is called the receptive field of the neuron, and this set of weights is called the convolution kernel. By performing the convolution calculation on the input feature map and the convolution kernel and, then, transferring the result to the nonlinear activation function, the next layer feature map is generated. The convolutional layer uses different convolution kernels to generate different feature maps. A single feature map is calculated by the same convolution kernel, which is called weight sharing. Weight sharing can reduce the complexity of the model and make the network easier to train. The forward propagation of the convolutional neural network from layer $l-1$ to layer l can be expressed by the following formula [11]:

$$x_j^l = f\left(\sum_{i\in M_j} x_i^{l-1}\cdot k_{ij}^l + b_j^l\right), \tag{1}$$

where x_j^l represents the output of the layer l, M_j represents the selected feature map, x_i^{l-1} represents the output of the layer $l-1$, k_{ij}^l represents the weight of layer l, and b_j^l represents the bias of layer l.

2.2. Activation Layer. The activation function is usually used to implement a nonlinear transformation on the output of convolution calculation to obtain a nonlinear representation of input data, thereby improving the feature-learning ability of the network. The activation function commonly used in the CNN is the Rectified Linear Unit (ReLU) function, and its calculation formula is [12]

$$f(x)\max(0,x)_{\text{cov}}, \tag{2}$$

where x is the input of the activation function.

To improve the model's antinoise ability, we use the Exponential Linear Unit (ELU) activation function in the multiscale convolution structure, which can speed up the learning process and improve the accuracy of the network. Similar to the ReLU function, the ELU avoids the problem of gradient disappearance by setting the positive part of the

input to be identical. But unlike the ReLU, the ELU does not set the negative value to zero, which is beneficial to speed up the network's learning speed. Also, it uses a saturation function in the negative part to make the ELU more robust to noise [13]. Its calculation formula is [13]

$$y_i = \begin{cases} z_i, & z_i \geq 0, \\ a\left(\exp\left(z_i\right) - 1\right), & z_i < 0, \end{cases} \tag{3}$$

where y_i is the activation function output value, z_i is the activation function input value, and a is a predefined parameter used to control the saturation value of the ELU for the negative input.

2.3. Pooling Layer. In the structure of convolutional neural networks, pooling layers are usually inserted between successive convolutional layers. Their role is to gradually reduce the dimension of the convolutional layer's output to reduce the parameters and calculations in the network and suppress overfitting and implement secondary feature extraction. The pooling layer is composed of multiple feature maps, and its feature maps correspond to the feature maps of the previous convolutional layer one by one without changing the number. The most commonly used pooling methods are maximum pooling and mean pooling. In this paper, the maximum pooling method is used because the performance of maximum pooling in one-dimensional time series tasks is better than that of average pooling [14]. Its calculation formula is [10]

$$P_i^{l+1}(j) = \max_{(j-1)W+1 \leq t \leq jW} \left\{q_i^l(t)\right\}, \tag{4}$$

where $q_i^l(t)$ represents the output of the tth neuron in the ith feature map of the layer l, $t \in [(j-1)W+1, jW]$, $W$is the width of the pooled area, and $P_i^{l+1}(j)$ is the pooled value of the corresponding neuron in the layer $l+1$.

3. Proposed MSK-1DCNN Model

3.1. Multiscale Convolution Structure. For the time series classification tasks using a one-dimensional convolutional neural network, the size of the convolution kernel has a significant impact on the performance of the network because part of the noise in the time series cannot be removed by BN, Bias, ReLU, and other operators. It can only be eliminated by the convolution operation of the convolution kernel [15]. The traditional one-dimensional convolutional neural network treats the size of the convolution kernel as a hyperparameter. It uses a fixed-size convolution kernel in convolution layer, which makes the design of the convolution kernel size a challenging problem. Also, the use of this method in prediction and classification tasks is limited because of the following problems: (1) Large-scale convolution kernels tend to focus on low-frequency regions and have good frequency resolution, but there are not enough convolution kernels in high-frequency regions, thereby ignoring high-frequency information. In contrast, the small-scale convolution kernel focuses on the frequency band, but the frequency resolution is low. (2) Using convolution kernels of the same size cannot adequately extract different discriminant features in the original signal [16]. To solve the abovementioned problems, scholars have proposed multiscale convolution. Multiscale convolution uses multiple filter banks of different scales to extract features from the original signal. It has been successfully applied in many fields, such as Environmental Sound Classification [17] and Speech Recognition [18]. Inspired by their work, we designed a two-layer multiscale kernel feature extraction structure (MSK), as shown in Figure 1.

In the first layer, MSK uses convolution kernels with a width of 11, 53, and 113 (the number is 16) to extract features from the original data to obtain three different feature maps and, then, stitch the three feature maps together as the output of the first layer. The second layer uses convolution kernels with a width of 36 and 72 (the number is 32) to continue extracting features from the first layer's output. Then, the convolution calculation results are stitched together and, finally, through the BN and ELU activation function layer to get the output feature map of MSK.

3.2. Network Structure and Parameters of MSK-1DCNN. In addition to the multiscale convolution structure, the proposed MSK-1DCNN has made the following improvements to the antinoise problem:

(1) The BN layer is added after the convolution layer. The BN layer is usually used before the activation function of the convolutional layer to readjust the data distribution. Its implementation steps are described in Algorithm 1 [19]. In a sense, γ and β represent the variance and offset of the input data distribution. For a network without BN, these two values are related to the nonlinear properties of the network's previous layer. After transformation, it is not associated with the previous layer. It becomes a learning parameter of the current layer, which is more conducive to optimization and does not reduce the network's ability [20]. BN can reduce the offset of covariance within input data, speed up the training process of deep neural networks, and reduce the dependence of the network on parameter initialization and increase the generalization ability.

(2) The ELU activation function is used instead of the ReLU function in the MSK structure. The ELU function avoids the problem of gradient disappearance and is more robust to noise, improving the network's antinoise ability.

The structure and parameters of MSK-1DCNN are shown in Figure 2.

The input of the network is a standardized fault vibration timing signal. The network directly extracts features from the original signal without any other signal processing. The MSK-1DCNN model consists of a feature extraction layer and a classification layer. The front of the feature extraction layer is a two-layer multiscale convolution structure, followed by a three-layer single-convolution kernel

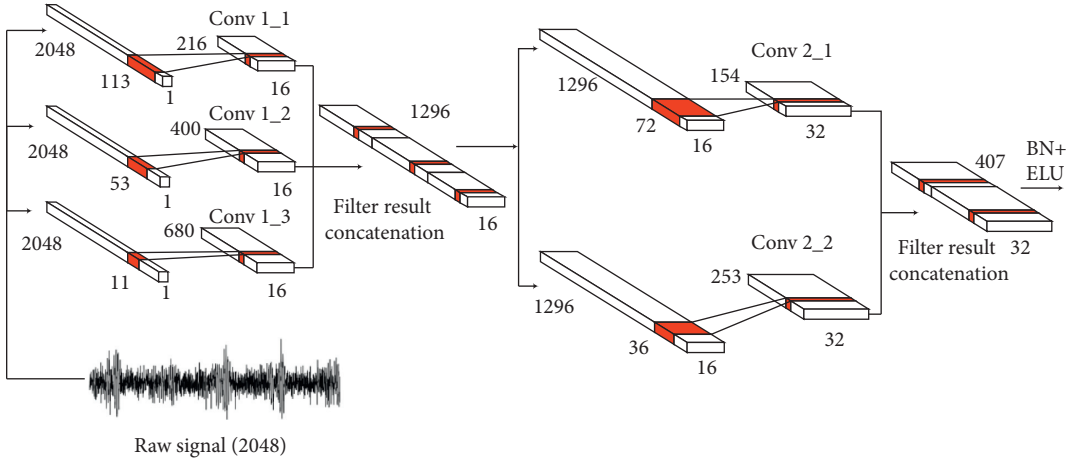


Figure 1: Multiscale convolutional structure.

Input: value of x over a mini-batch: $B = \{x_1, \ldots m\}$; Parameters to be learned: γ, β
Output: $\{y_i = BN_{\gamma,\beta}(x_i)\}$
$\mu_B = (1/m)\sum_{i=1}^{m} x_i$//mini-batch mean
$\sigma_B^2 = (1/m)\sum_{i=1}^{m} (x_i - \mu_B)^2$//mini-batch variance
$\widehat{x}_i = ((x_i - \mu_B)/(\sqrt{\sigma_B^2 + \varepsilon}))$//normalize
$y_i = \gamma\widehat{x}_i + \beta = BN_{\gamma,\beta}(x_i)$//scale and shift

Algorithm 1: BN.

convolution layer. The multiscale convolution structure can extract different discriminative features from the original signal. The single-convolution kernel convolution layer can extract more advanced features and deepen the depth of the network to improve the antinoise ability of the model. The classification layer is composed of two fully connected layers and a softmax layer. The softmax function is used to convert the network output into a probability distribution form that conforms to the bearing's ten fault states. The formula of softmax is as follows:

$$q(z_j) = \frac{e^{z_j}}{\sum k^{10} e^{z_k}}, \tag{5}$$

where z_j represents the normalized probability of output of the jth neuron through the softmax function.

3.3. MSK-1DCNN Fault Diagnosis Model. MSK-1DCNN fault diagnosis model is established in three steps:

(1) Data preprocessing: the original data are cut into a data set according to the resampling method [21]. The data set is divided into a training set and a test set according to a certain ratio, and then, pepper noise is added to the training set. Then, the Gaussian white noise (simulating noise in daily industrial production) is added to the testing set.

(2) Training model: we select an Adam optimizer and cross-entropy loss function. The Adam algorithm is easy to implement and has high computational efficiency with low memory requirements [22]. Its optimization performance is better than that of the SDG and RMSprop optimizer. The cross-entropy function was chosen because it is an entropy-shaped loss function. It is insensitive to noise and suitable for intense noise environments [23]. The initial value of the learning rate is set to 0.0005 and decreases by 0.0001 every 10 iterations. It does not decrease until the learning rate is 0.0001 and is fixed at 0.0001. The model is trained on the training set until the loss value is fully converged, stops the iteration, and saves the trained model.

(3) Testing model: we use the testing set to test the model and take the average of five testing results as the final testing result.

4. Experiment

4.1. Data Set. The bearing data set is provided by the Western Reserve University Bearing Data Center (https://csegroups.case.edu/bearingdatacenter/home), and its fault test bench is shown in Figure 3.

We selected the motor drive end bearing data sampled at 48 kHz at 1hp, 2hp, and 3hp. The fault type includes normal, inner ring failure, outer ring failure, and roller failure. Each fault contains 3 fault levels with widths of 0.007, 0.014, and 0.021 inches, so the data set has ten states. According to the resampling method [22], the original data of ten states are divided. Each sample has a length of 2048 and a sampling step size of 480. Therefore, each state obtains 1000 samples at

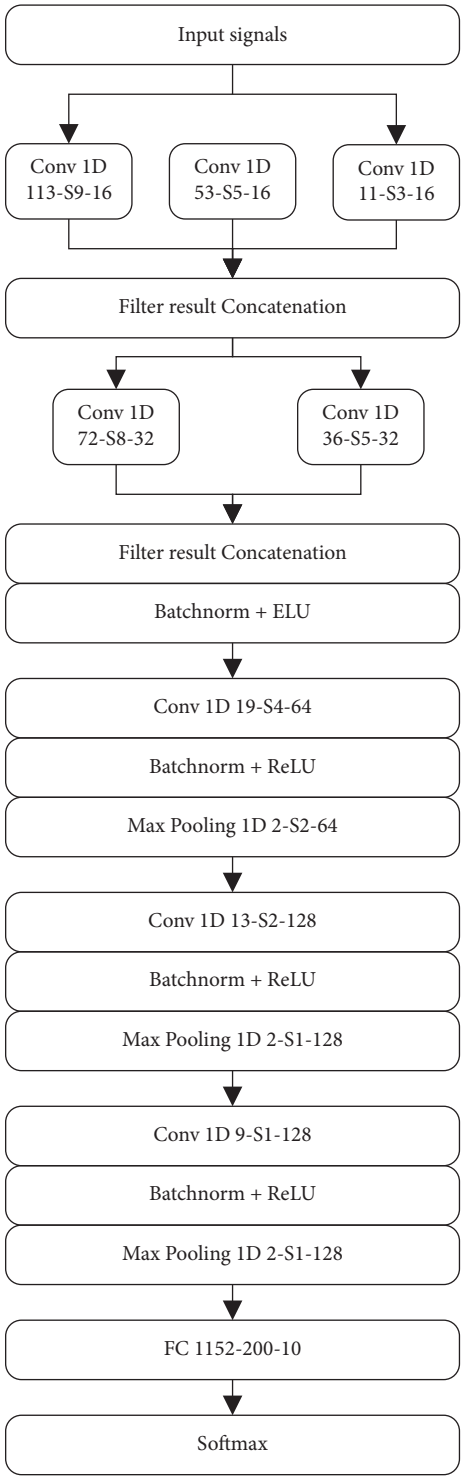


Figure 2: Architecture and parameters of MSK-1DCNN.

1 hp, 2 hp, and 3 hp, respectively, and a total of 30,000 samples in ten states of three loads constitute a data set. We divide 85% of the data set into a training set and add pepper noise, and 15% of the data set is divided into a test set and added with the Gaussian noise of different SNR. SNR is the signal noise ratio, representing the ratio of the original signal's power to the noise power, usually expressed in decibels. The smaller the value of SNR, the stronger the noise. The formula of SNR is as follows:

$$\mathrm{SNR}_{\mathrm{dB}} = 10 \log_{10}\left(\frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}}\right), \tag{6}$$

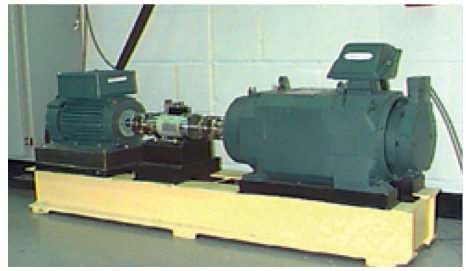


Figure 3: Motor driving mechanical system used by CWRU.

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

4.2. Effectiveness of the Multiscale Convolution Structure. To verify the proposed multiscale convolution structure's effectiveness, we compared MSK-1DCNN with a network using a single layer with a single-convolution kernel. The first layer of the single-convolution kernel network uses five convolution kernels (widths of 11, 36, 53, 72, and 113, respectively) in MSK, as shown in Figure 1, to extract features from the original signal. After that, the network structure is consistent with the network structure after the multiscale convolution structure of MSK-1DCNN. The specific parameters of the single-kernel convolution layer are shown in Table 1. To maintain the consistency of the rest of the network structure, we did padding for input data.

The experimental results are shown in Figure 4. It can be seen that compared with the single-layer and single-size convolution kernel, the accuracy of the network using a multiscale convolution structure at low SNR is significantly improved. It shows that the multiscale convolution structure considers both the high-frequency region and low-frequency region and can extract various discriminative features with different resolutions from the original signal and is more robust to noise. Simultaneously, it can also be found that the diagnostic accuracy of single-convolution kernel networks of various sizes is different, indicating that the size of the convolution kernel will have a significant impact on the network diagnostic accuracy and also reflecting the importance of convolution kernel design.

4.3. Effect of Pepper Noise on the Performance of Network Feature Extraction. Salt and pepper noise, also known as impulse noise, is a white- and dark-spot noise generated by the image sensor, transmission channel, and decoding processing in the image. Pepper is a black point (pixel value 0), and salt is a white point (pixel value 225). Generally, pepper and salt noise are added by randomly changing some pixel values to 0 or 225. When training a Denoising Auto Encoder (DAE), pepper noise is usually added to the input training data to improve the autoencoder's feature extraction capability. This method is realized by randomly setting the input data to 0 with a certain probability, that is, dropout the input data with a certain probability [24]. Because of this method's successful application in the DAE, we add pepper noise to training data by randomly zeroing it with a probability of 0.5.

We compared the effects of adding pepper noise and not adding pepper noise to training data on network feature extraction ability (the other structures and parameters remain the same) through experiments. The results are shown in Table 2. It can be seen that the accuracy of the network with noisy training is higher than that without noise when the SNR is low. It shows that adding pepper noise in the training set can effectively destroy the original data, make the network learn more essential features of input data, suppress overfitting, and improve the accuracy of the network under noise.

4.4. Activation Function. The ReLU is the most widely used activation function in neural networks. Its simple operation of taking the maximum value makes its calculation speed much faster than that of the Sigmoid or tanh activation function. It also caused the sparsity in the hidden units, and there was no problem of gradient disappearance. But, its operation of zeroing negative values will cause the death of neurons and is not robust to noise. The ELU function retains the advantages of the ReLU function. Instead of zeroing the negative value part, the saturation function is used in the negative value part, making the ELU more robust to noise.

We use the ELU activation function in the MSK and compare it with MSK using the ReLU through experiments. Except for the different activation functions of two networks, the other structures and parameters remain the same. The results are shown in Figure 5. It can be found that the ELU function performs better than the ReLU function at low SNR. It is proved that the ELU activation function is insensitive to noise and has intense antinoise ability under a strong noise environment. Therefore, the MSK-1DCNN using the ELU function achieves better diagnostic performance at low SNR.

4.5. Comparison of Fault Diagnosis Accuracy. To verify the effectiveness of the proposed MSK-1DCNN, the WDCNN [10] proposed by Zhang et al., Stacked Autoencoder, and BP neural network are used as comparative models for experiments. WDCNN uses the wide convolution kernel in the first convolution layer to extract features and suppress high-frequency noise. It uses the small convolution kernel in the few layers for multilayer nonlinear mapping and uses the AdaBN algorithm to improve the domain adaptation ability.

Table 1: Parameters of the single-scale convolutional structure.

Kernel size	Stride	Kernel number	Padding
11	5	32	0
36	5	32	9
53	5	32	18
72	5	32	27
113	5	32	48

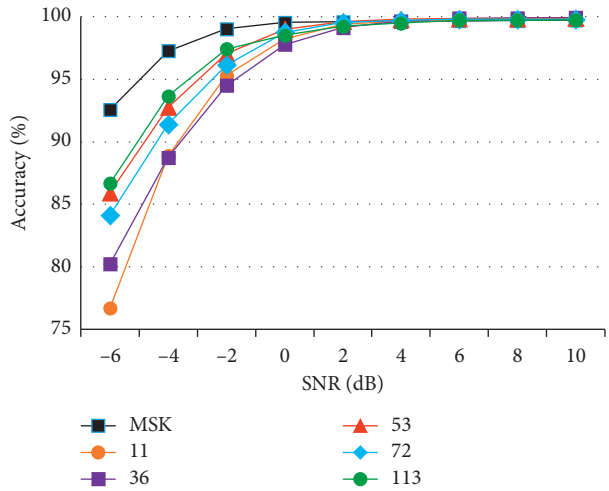


Figure 4: Diagnosis accuracy of the multiscale CNN and single CNN model.

Table 2: Diagnosis accuracy of noise and no-noise input training models.

SNR/db	−6	−4	−2	0	2	4	6	8	10
With noise	92.55%	97.27%	99.03%	99.53%	99.58%%	99.64%	99.60%	99.68%	99.72%
With no noise	61.81%	78.17%	89.01%	95.43%	98.17%%	99.19%	99.64%	99.79%	99.87%

The diagnostic accuracy of WDCCNN at low SNR is significantly higher than that of other convolution models proposed in recent years, such as the TICNN [6] proposed in their another paper and the ACNNDM-1D [25] proposed by Liu et al. SAE is formed by stacking three autoencoders, the number of neurons in the middle layer is 800, 200, and 50 respectively, and the Sigmoid activation function and the MSEloss function are used. When training SAE, 5% of the training set is divided into the validation set to fine tune the SAE. The specific training method is carried out according to the method in [5]. The number of neurons in the BP neural network is 2048, 1000, 500, 200, and 10, and the Sigmoid activation function and the cross-entropy loss function are used.

Table 3 lists the experimental results of the four models on the no-noise data set. It can be seen that their accuracy on the training set has reached 100%, showing the strong fitting ability of the deep neural network. The test accuracy of MSK-1DCNN and WDCNN reached 100%, while BP and SAE's test accuracy was less than 100%. It shows that compared to the fully connected structure, the network using convolution calculation has a stronger ability to suppress overfitting.

It can be seen from Figure 6 that the diagnostic accuracy of the convolutional structure is significantly higher than the SAE and BP neural network because the convolutional structure has more advantages in processing one-dimensional time series data and has stronger noise resistance. BP neural network and SAE's full connection structure leads to serious network overfitting, so even if the SNR is high, the diagnostic accuracy of them is low. The diagnosis accuracy of the MSK-1DCNN fault diagnosis model we proposed at low SNR is significantly higher than that of other models, which proves the effectiveness of improvements made in this paper of bearing faults diagnosis under the noise environment.

To further demonstrate the performance of the proposed MSK-1DCNN, we use the t-SNE (t-distributed stochastic neighbor embedding) algorithm to visualize the output of each model's last layer mentioned above. The output dimension of each model on the testing set is reduced using the t-SNE algorithm when the SNR is −4, and the results are displayed in a two-dimensional space, as shown in Figure 7.

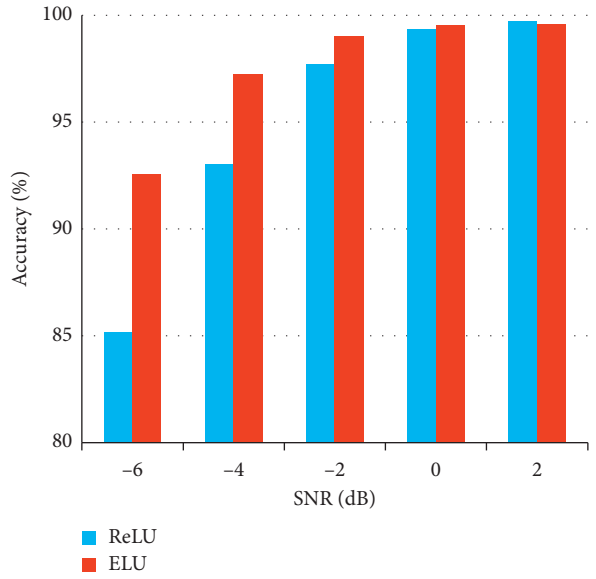
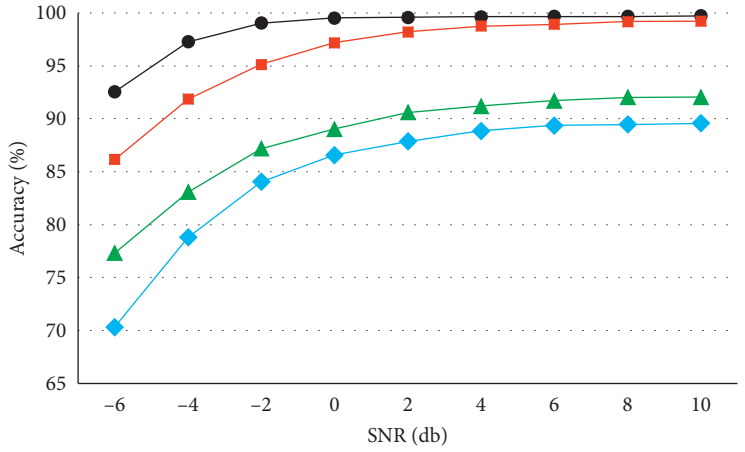


Figure 5: Diagnosis accuracy of MSK-1DCNN with an ReLU or ELU.

Table 3: Diagnosis accuracy of each model on the no-noise dataset.

Model	SAE (%)	BP (%)	MSK-1DCNN (%)	WDCNN (%)
Train Acc	100	100	100	100
Test Acc	97.12	98.77	100	100



MSK-1DCNN	92.55	97.27	99.03	99.53	99.58	99.64	99.66	99.68	99.72
WDCNN	86.15	91.85	95.15	97.19	98.23	98.76	98.93	99.2	99.22
MLP	77.32	83.08	87.17	89.04	90.61	91.19	91.72	92.02	92.06
SAE	70.34	78.8	84.06	86.59	87.86	88.86	89.37	89.46	89.57

Figure 6: Comparison of diagnosis accuracy.

The original input signal's fault state is chaotic and inseparable, for the features are gathered into a distinguishable state after the feature extraction of each model. It can be seen that the output features of SAE are poorly aggregated. Although the various features of the BP neural network are gathered together, the features overlap with each other and are not completely separated. The outputs of WDCNN and MSK-1DCNN have only few fault states overlapping and

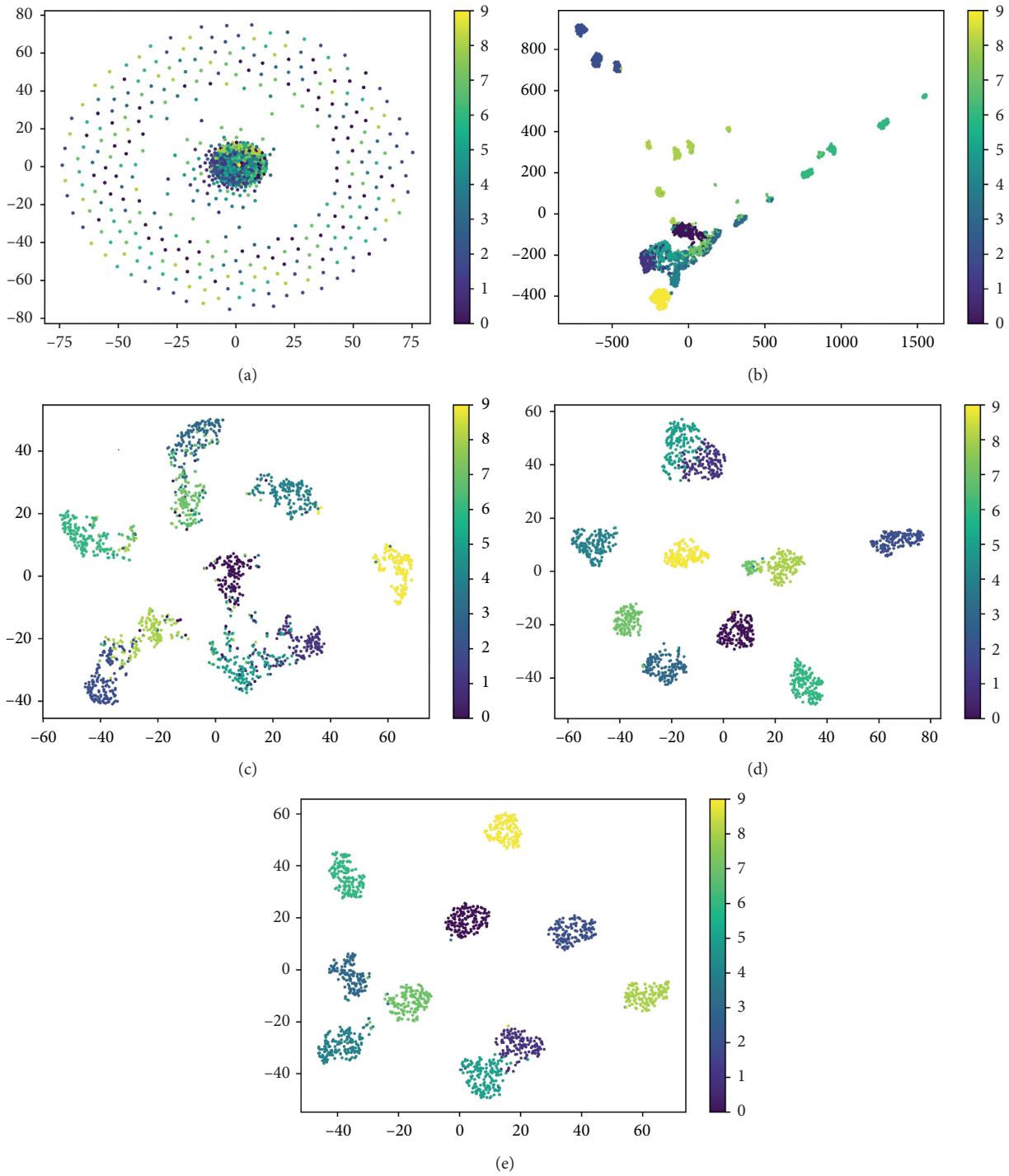


Figure 7: Visualization of the input extracted from the last layer of different test models via the t-SNE method: (a) raw input signals; (b) SAE; (c) BP Neural Network; (d) WDCNN; and (e) MSK-1DCNN.

achieve separation of each fault state. It shows that these two models based on the one-dimensional convolutional neural network have more robust feature extraction capabilities. Moreover, the feature aggregation degree of MSK-1DCNN is better than that of WDCNN, which proves that the diagnostic performance of the MSK-1DCNN model under intense noise is better than that of WDCNN.

5. Conclusions

To solve the impact of the noise collected by the sensor on the diagnostic accuracy of the convolutional neural network, we propose a one-dimensional convolutional neural network with multiscale convolution kernels (MSK-1DCNN) in this paper. MSK-1DCNN uses a multiscale convolution

structure to extract different fault features from the original signal and use the ELU function instead of the ReLU function in the MSK structure. At the same time, we use a training set with pepper noise to train MSK-1DCNN. The experimental results show that the MSK structure can extract discriminative features with different resolutions from the original signal, ELU activation function can effectively improve the antinoise ability, and adding pepper noise to training data during the training stage of the network can make the network more robust. The diagnostic accuracy of MSK-1DCNN at low SNR is significantly higher than that of other comparison models, which shows the powerful antinoise ability of MSK-1DCNN.

The number of samples of each fault type in this paper is entirely balanced, but the actual industrial environment's data are not entirely balanced. The imbalance of the sample number will significantly affect the accuracy of the model classification results. Therefore, we consider solving the problem of noisy fault diagnosis of the bearing under the imbalanced data set in the future work.

Data Availability

The bearing data set is provided by the Western Reserve University Bearing Data Center (https://csegroups.case.edu/bearingdatacenter/home).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The present work was funded by the National Natural Science Foundation of China (Grant nos. 61763028 and 62063020) and the Natural Science Foundation of Gansu, China (Grant no. 20JR5RA463).

References

[1] D.-T. Hoang and H.-J. Kang, "A survey on deep learning based bearing fault diagnosis," *Neurocomputing*, vol. 335, pp. 327–335, 2019.

[2] C. Szegedy, S. Ioffe, V. Vanhoucke et al., "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016, https://arxiv.org/abs/1602.07261.

[3] T. Mikolov, M. Karafiát, L. Burget et al., "Recurrent neural network based language model," *Interspeech*, 2010.

[4] H. Greenspan, B. van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: overview and future promise of an exciting new technique," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1153–1159, 2016.

[5] M. Xia, T. Li, L. Liu, L. Xu, and C. W. de Silva, "Intelligent fault diagnosis approach with unsupervised feature learning by stacked denoising autoencoder," *IET Science, Measurement & Technology*, vol. 11, no. 6, pp. 687–695, 2017.

[6] W. Zhang, Li Chuanhao, P. Gaoliang et al., "A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load," *Mechanical Systems and Signal Processing*, vol. 100, pp. 439–453, 2018.

[7] H. Shao, H. Jiang, F. Wang, and H. Zhao, "An enhancement deep feature fusion method for rotating machinery fault diagnosis," *Knowledge-Based Systems*, vol. 119, pp. 200–220, 2017.

[8] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time motor fault detection by 1-d convolutional neural networks," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, 2016.

[9] L. Eren, "Bearing Fault detection by one-dimensional convolutional neural networks," *Mathematical Problems in Engineering*, vol. 2017, Article ID 8617315, 9 pages, 2017.

[10] W. Zhang, G. Peng, C. Li et al., "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors*, vol. 17, no. 3, p. 425, 2017.

[11] J. Bouvrie, *Notes on Convolutional Neural Networks*, MIT CBCL Tech Report, Cambridge, MA, USA, 2006.

[12] V. Nair, G. E. Hinton, and C. Farabet, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning*, pp. 807–814, Haifa, Israel, June 2010.

[13] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," 2015, https://arxiv.org/abs/1511.07289.

[14] S. Huang, J. Tang, J. Dai, Y. Wang, and J. Dong, "1DCNN fault diagnosis based on cubic spline interpolation pooling," *Shock and Vibration*, vol. 2020, Article ID 1949863, 13 pages, 2020.

[15] W. Tang, G. Long, L. Liu et al., "Rethinking 1D-CNN for time series classification: a stronger baseline," 2020, https://arxiv.org/abs/2002.10061.

[16] Y. Yao, S. Zhang, S. Yang, and G. Gui, "Learning attention representation with a multi-scale CNN for gear fault diagnosis under different working conditions," *Sensors*, vol. 20, no. 4, p. 1233, 2020.

[17] B. Zhu, C. Wang, F. Liu et al., "Learning environment sounds with multi-scale convolution neural network," 2018, https://arxiv.org/abs/1803.10219.

[18] Z. Zhu, J. Engel, and A. Hannun, "Learning multiscale feature directly from waveforms," 2016, https://arxiv.org/abs/1603.09509.

[19] I. Sergey and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," 2015, https://arxiv.org/abs/1502.03167.

[20] J. Bjorck, C. Gomes, and B. Selman, "Understanding batch normalization," 2018, https://arxiv.org/abs/1806.02375.

[21] C. Li, W. Zhang, G. Peng, and S. Liu, "Bearing Fault diagnosis using fully-connected winner-take-all autoencoder," *IEEE Access*, vol. 6, pp. 6103–6115, 2018.

[22] D. Kingma and B. Jimmy, "Adam: a method for stochastic optimization," 2015, https://arxiv.org/abs/1412.6980.

[23] H. Shao, H. Jiang, H. Zhao, and F. Wang, "A novel deep autoencoder feature learning method for rotating machinery fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 95, pp. 187–204, 2017.

[24] 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 Processing*, vol. 130, pp. 377–388, 2017.

[25] X. Liu, Q. Zhou, J. Zhao et al., "Real-time and anti-noise fault diagnosis algorithm based on 1-D convolutional neural network," *Journal of Harbin Institute of Technology*, vol. 51, no. 7, pp. 89–95, 2019.