Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 2604191, 16 pages
https://doi.org/10.1155/2020/2604191

Research Article

Wasserstein Generative Adversarial Network and Convolutional Neural Network (WG-CNN) for Bearing Fault Diagnosis

Hang Yin,[1,2] **Zhongzhi Li**,[2] **Jiankai Zuo**,[3] **Hedan Liu**,[2] **Kang Yang**,[4] **and Fei Li**[2]

[1]*School of Information Technology, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China*
[2]*School of Computer Science, Shenyang Aerospace University, Shenyang 110136, China*
[3]*Department of Computer Science and Technology, Tongji University, Shanghai 201804, China*
[4]*Liaoning General Aviation Academy, Shenyang 110000, China*

Correspondence should be addressed to Hang Yin; yinhang@sau.edu.cn and Hedan Liu; liuhedan33@sina.com

Received 1 February 2020; Revised 7 April 2020; Accepted 22 April 2020; Published 11 May 2020

Academic Editor: Alessandro Gasparetto

In recent years, intelligent fault diagnosis technology with deep learning algorithms has been widely used in industry, and they have achieved gratifying results. Most of these methods require large amount of training data. However, in actual industrial systems, it is difficult to obtain enough and balanced sample data, which pose challenges in fault identification and classification. In order to solve the problems, this paper proposes a data generation strategy based on Wasserstein generative adversarial network and convolutional neural network (WG-CNN), which uses generator and discriminator to conduct confrontation training, expands a small sample set into a high-quality dataset, and uses one-dimensional convolutional neural network (1D-CNN) to learn sample characteristics and classify different fault types. Experimental results over the standard Case Western Reserve University (CWRU) bearing fault diagnosis benchmark dataset showed that the proposed method has obvious and satisfactory fault diagnosis effect with 100% classification accuracy for few-shot learning. In different noise environments, this method also has excellent performance.

1. Introduction

In recent years, with the rapid development of high-performance computing, big data, and deep learning technologies, intelligent fault diagnosis and pattern recognition technology based on deep learning has attracted wide attention from scholars in the field because it does not rely on subjective human analysis and reasoning. There are many research studies on fault diagnosis based on open source data, which can better reflect the comparative and dynamic nature of the research. There are basically four major open source bearing fault datasets in the world, Case Western Reserve University (CWRU) datasets, Paderborn University bearing datasets, PRONOSTIA bearing dataset, and Intelligent Maintenance Systems (IMS) datasets.

The application of the earliest artificial neural network (ANN) algorithms to motor faults can be traced back to [1, 2], and ANN was used to make a comprehensive induction of different types of motor faults. Schoen et al., Li et al., Cococcioni et al., and Qian et al. [3–6] all used a certain degree of human experience knowledge to make fault feature selection to train ANN more effectively. In [7], it was one of the earliest literatures on bearing fault diagnosis using PCA. In addition, classic papers based on PCA [8, 9] made use of their data mining capabilities to promote "manual" feature selection and extract more representative fault features. In the problem of bearing fault classification, the results obtained by SVM in [10] are the best in all cases, which comprehensively improved the performance of ANN. Other papers [11–13] used KNN to conduct distance analysis on the new data sample and determine whether it belonged to a specific fault category. In addition to the aforementioned common ML methods, many other classification algorithms have also been applied to the identification of bearing faults, Bayesian networks [14], ELM [15], transfer learning [16], random forest [17], independent component analysis [18], manifold learning [19], typical variable analysis [20], expectation maximization [21], set learning [22], empirical model decomposition [23], and dictionary learning [24].

However, the industrial environment is more complex; some bearing fault characteristics are difficult to be artificially extracted or explained because of high dimensional characteristics. These "weak" classical machine learning methods based on manually selected features sometimes give inaccurate classification results. Therefore, many deep learning algorithms with automatic feature extraction capability and better classification performance have been successfully applied to bearing fault diagnosis and also achieved gratifying results.

The first paper using CNN to identify bearing faults [25] was published in 2016. In the following three years, papers using the same technology [26–30] promoted the development of various bearing fault detection. Additionally, other deep neural networks have been successfully applied in this field, such as deep belief network (DBN) [31], recurrent neural network (RNN) [32], autoencoders [33], generative adversarial network (GAN) [34], and other related technologies. Among them, GAN was proposed in 2014 and quickly became one of the most exciting breakthroughs in the field of deep learning. In order to achieve a better balance between training speed and accuracy, adaptive CNN (ADCNN) was applied to the CWRU dataset to change the learning rate dynamically in [26]. Many variants of CNN have also been used to solve bearing fault diagnosis [27–30]. Semisupervised generative adversarial network (SSGAN) was used on the CWRU dataset to achieve a gratifying result [35].

Although the above studies have achieved encouraging results, there are still some problems such as difficulty in data collection and a large amount of noise in the data in bearing fault detection field. When using ML and DL algorithms, sufficient and effective training data samples cannot be obtained, so it is difficult to achieve extremely high classification accuracy. This paper proposes a deep neural network based on the combination of GAN and CNN to solve the problem of bearing fault diagnosis with limited data. This model greatly improves the accuracy of classification and the robustness of the model in the case of reducing the dependence on the original dataset.

Our contributions mainly include the following:

(1) This paper proposes a pure data-driven method based on WGAN to artificially synthesize new annotated fault type samples. More specifically, we use WGAN to estimate the distribution of observed fault samples and generate new samples that can be used for training deep networks. With this strategy, our training dataset is expanded and enhanced. One-dimensional convolutional neural network (1D-CNN) is used to extract classification features efficiently and to train the model from original and generated samples.

(2) We analyzed the difference between the original data and the generated data from qualitative and quantitative perspectives.

(3) In order to effectively improve the classification effect of the CNN model, we optimized the relevant quantities (convolution kernel settings, activation function, batch size, and learning rate).

(4) Compared with other comparative experiments, the proposed model was verified on the CWRU bearing fault diagnosis benchmark dataset with an accuracy of 100%.

The other parts of the paper are organized as follows. Section 2 introduces the basic model of CNN and WGAN and the proposed model based on WG-CNN. Section 3 presents the experimental process and a certain result analysis. Section 4 summarizes this paper.

2. Models and Methods

2.1. GAN and WGAN Models. The generative adversarial network (GAN) consists mainly of two submodules: the generator model is defined as G and the discriminator model is defined as D. GAN is based on the idea of competition. The purpose of G is to confuse D, and the purpose of D is to distinguish between the generated data from G and the data from the original dataset.

In more detail, G: $Z \longrightarrow X$, where Z is the noise space which has any dimensions; it is corresponding to the hyperparameter space. X is the data space; the purpose is to get the data distribution. The generator generates new data by fitting data features in the data space and randomly adding noise. The value function of G and D is

$$\min_{\theta} \max_{\omega} E_{x\sim P_r}\left[\log D_{\omega}(x)\right] - E_{z\sim P_z}\left[\log D\left(g_{\theta}(z)\right)\right]. \tag{1}$$

However, the above methods have a series of problems such as unstable training and unconvergent generator loss function. WGAN [36] solved the problem of GAN training instability. It uses the Wasserstein distance as shown in the following equation:

$$W\left(P_r, P_{\theta}\right) = \inf_{\gamma\sim\prod\left(P_r,P_{\theta}\right)} E_{(x,y)\sim\gamma}\left[\|x-y\|\right], \tag{2}$$

where P_r and P_{θ} are the distribution of the original data and generated data and $\prod$ $(P_r,$ $P_{\theta})$ represents the joint distribution.

$$\begin{aligned} W\left(P_r, P_{\theta}\right) &= \sup_{\|f\|\leq 1} E_{x\sim P_r}\left[f(x)\right] - E_{x\sim P_{\theta}}\left[f(x)\right] \\ &= \frac{1}{K}\sup_{\|f\|_L\leq K} E_{x\sim P_r}\left[f(x)\right] - E_{x\sim P_{\theta}}\left[f(x)\right] \\ &\Longrightarrow \max_{\|f\|_L\leq 1} E_{x\sim P,P}\left[f(x)\right] - E_{x\sim P_{\theta}}\left[f(x)\right]. \end{aligned} \tag{3}$$

For the aforementioned equation, the loss function of WGAN is

$$\min_{\theta} \max_{\omega} E_{x\sim P_r}\left[f_{\omega}(x)\right] - E_{z\sim P_z}\left[f\left(g_{\theta}(z)\right)\right]. \tag{4}$$

Compared with GAN, WGAN almost solves the problem of unstable training, thus ensuring the diversity of generated samples. In this paper, we use the WGAN proposed in [36] as the data generation model and adopt the gradient penalty strategy of [37]. The overall structure of the generator and discriminator is shown in Figure 1. And the network structure of generator and discriminator is the same, as shown in Figure 2.

2.2. CNN Model. The convolutional neural network is a typical deep feed-forward neural network with a structure similar to that of a multilayer perceptron. Unlike multilayer perceptrons, CNN contains a convolution kernel for extracting features. In a convolutional layer, there are usually several feature planes; each of them consists of a number of rectangular arranged neurons, and one neuron is only connected to a part of the adjacent layer neurons. Its advantage is reducing network complexity, improving computing efficiency, and increasing the network's ability to fit. A typical CNN architecture consists of the following basic elements.

2.2.1. Convolutional Layer. The convolutional layer consists of several convolutional neurons. The weight and parameters of each neuron can be derived by a backpropagation algorithm. A convolution filter within the convolutional layer provides a compressed representation of the input data. Convolution filters can extract features from the input data. Each filter consists of weights that are adjusted during the training phase of the network. Through the convolutional layer, the network can extract low-level edge features of the input data vector. The calculation formula of the convolutional layer is as follows:

$$x_m^k = f\left(\sum_{j=m}^{n} x_j^{k-1} w_j^k + b_j^k\right), \tag{5}$$

where ${x^k}_m$ and ${x^{k-1}}_m$, respectively, represent the outputs of the m^{th} node in the k^{th} layer and the $k-1^{th}$ layer and w_j^k and ${b^k}_j$, respectively, represent the weights and thresholds corresponding to the m^{th} node in the k^{th} layer.

2.2.2. Pooling Layer. The pooling layer neurons perform pooling operations on the features obtained by the convolutional layer, thereby segmenting the feature regions to obtain features with smaller values such as the highest value and the mean value of the feature regions. The transformed features are subsampled by a particular factor in the subsampling layer. In order to calculate the values of a particular feature in an area of the input layer and merge them together, the role of the subsampling layer is to reduce the variance of the transformed data. The expression of the pooling layer is as follows:

$$x_m^k = h\left(x_m^{k-1}\right) + b_m^k, \tag{6}$$

where $h(x)$ represents the maxpooling function.

2.2.3. Fully Connected Layer. The fully connected layer converts all local features into global features, thereby obtaining the output of the network. The neuron can be expressed as follows:

$$y = f\left(w^{k-1} x^{k-1} + b^k\right), \tag{7}$$

where y represents the output of the neuron and w^{k-1} and x^{k-1} represent the connection weights and thresholds of the k^{th} layer to the $k-1^{th}$ layer, respectively.

2.2.4. Activation Function. After several convolutional and pooling layers, the classification function in the neural network is performed through fully connected layers. The neurons of the fully connected layer are all connected to all activation functions in the previous layer. The fully connected layer ultimately transforms the two-dimensional feature map to a one-dimensional feature vector. The derived vector can be used for two or more classifications and can be used for further processing. The structure of CNN is shown in Figure 3.

2.3. The Proposed Model. In order to solve the problem of limited training data, this paper proposes a classifier method combining Wasserstein GAN and CNN. Firstly, the original sample is divided into training set samples and test set samples, and the generated training network is used to enhance the data of the fault training samples to generate a large number of simulated samples. Then, the samples mixed by generated samples and original samples are used to training the deep learning classifier based on CNN. Finally, the trained classifier is tested using test samples to verify the effectiveness of the method for the limited data problem.

2.3.1. Input of WG-CNN. The input data are derived from the drive end bearing data. We directly use WGAN to extend the original dataset. The data are expanded as the input of the CNN classifier, and we use the CNN classifier for direct feature extraction. This method is called an end-to-end learning training process.

2.3.2. Output of WG-CNN. The output of WGAN is a generated dataset that is consistent with the shape and size of the original input dataset. The fusion dataset is input into the CNN classifier through fusion with the original data and the generated data. The composite dataset is reduced by the feature extraction of the convolution kernel and the feature dimension of the pooling layer. After the fully connected layer, the classification is performed using the activation function (8). Finally, the final classification situation is determined by the output of each class score.

$$y_i = \frac{e^{a_i}}{\sum_{i=1}^{C} e^{a_k}}. \tag{8}$$

2.3.3. WG-CNN Loss Functions. Wasserstein distance is given in equation (2). The loss functions of the generator and discriminator are (9) and (10), respectively.

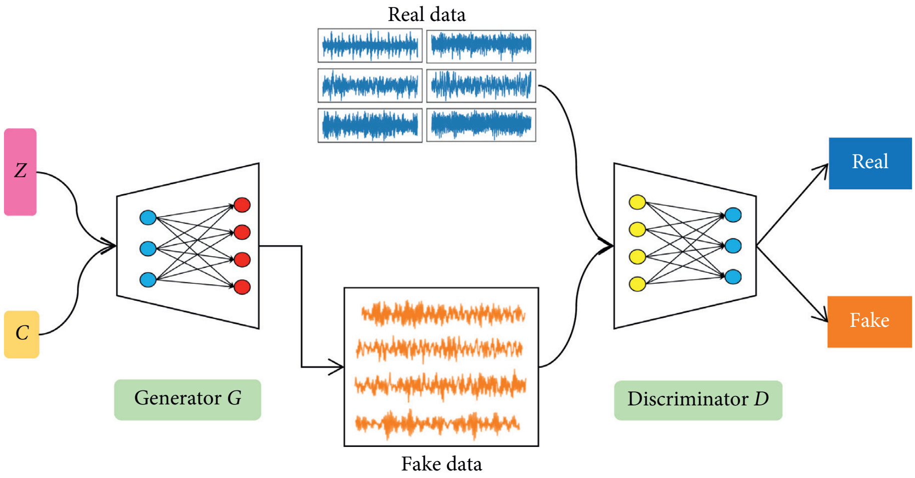


Figure 1: WGAN framework.

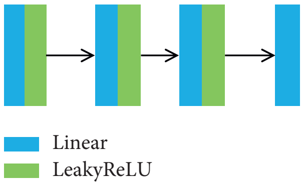


Figure 2: Generator and discriminator's network structure.

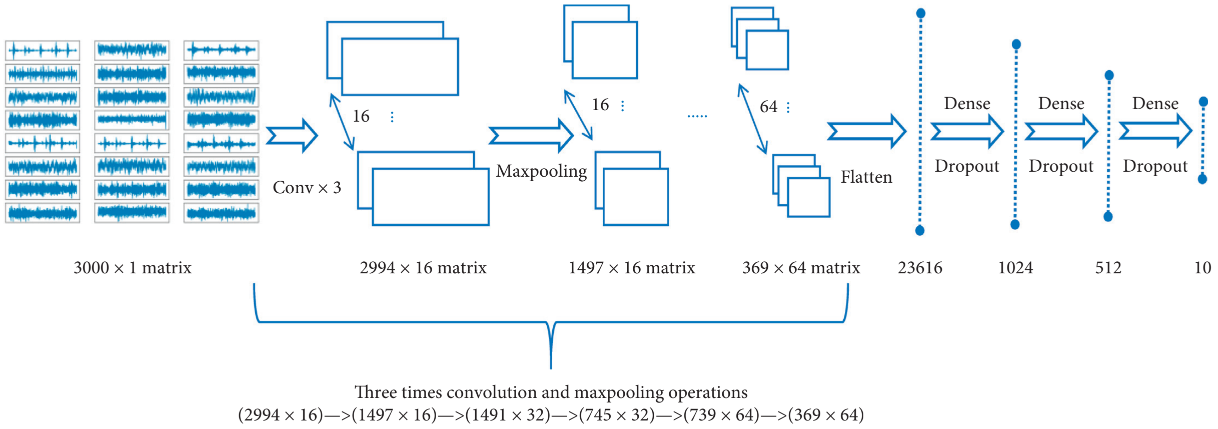


Figure 3: CNN framework.

$$L(G) = -E_{x \sim P_\theta}[D(x)], \tag{9}$$

$$L(D) = -E_{x \sim P_r}[D(x)] + E_{x \sim P_\theta}[D(x)], \tag{10}$$

where P_r and P_θ are the distribution of the original data and the generated data and $D(x)$ represents the output of the discriminator.

Due to the interaction between the weight constraint and the cost function, the WGAN optimization process is difficult, which can cause the gradient to disappear or explode. The main reason is that WGAN performs gradient truncation. Gradient truncation will result in the discriminating network tending to a binary network; it will cause a drop in model capacity. According to [37], it is pointed out that gradient penalty is used instead of gradient clipping, so the loss function of the discriminator is expressed as follows:

$$\begin{aligned} L(D) = {} & \underbrace{-E_{x \sim P_r}[D(x)] + E_{x \sim P_\theta}[D(x)]}_{\text{Original critic loss}} \\ & + \underbrace{\lambda E_{x \sim P_x}\left[\|\nabla_x D(x)\|_p - 1\right]^2}_{\text{Gradient penalty}}, \end{aligned} \tag{11}$$

where $[\|\nabla_x D(x)\|_p - 1]^2$ is the gradient penalty and λ is the gradient penalty weight parameter.

This problem is a typical multiclassification problem, so the loss function of the CNN classifier uses cross entropy loss. For sample point (x, y), y is the real label. In the multiclass problem, the value can only be the label set. We assume that there are K label values, and the probability that the i^{th} sample is predicted as the k^{th} label value is $P_{i,k}$, that is, $P_{i,k} = P_r(t_{i,k} = 1)$ has a total of N samples, and the loss function of the model is expressed as follows:

$$L_{\log}(Y, P) = -\log \Pr(Y|P) = -\frac{1}{N}\sum_{i=0}^{N-1}\sum_{k=0}^{K-1} y_{i,k}\log p_{i,k}. \quad (12)$$

2.3.4. WG-CNN Optimizers. The RMSProp algorithm is called Root Mean Square Prop, which is used as the optimizer of the WGAN model. In order to further optimize the problem that the loss function has too large swing amplitude during the process of update and further accelerate the convergence speed of the function, the RMSProp algorithm uses the differential squared weighted average for the gradient of the weight W and the offset b. In the t^{th} iteration, the formula is (13) to (16).

$$s_{d\omega} = \beta s_{d\omega} + (1 - \beta)\mathrm{d}W^2, \quad (13)$$

$$s_{db} = \beta s_{db} + (1 - \beta)\mathrm{d}b^2, \quad (14)$$

$$W = W - \alpha \frac{\mathrm{d}W}{\sqrt{s_{\mathrm{d}\omega} + \varepsilon}}, \quad (15)$$

$$b = b - \alpha \frac{\mathrm{d}b}{\sqrt{s_{\mathrm{d}b} + \varepsilon}}, \quad (16)$$

where α is the learning rate of the network, S_{dw} and $S_{\mathrm{d}b}$ are the gradient momentums accumulated by the loss function in the previous t−1 round iteration, and β is an index of the gradient accumulation. Unlike other optimizations, the RMSProp algorithm calculates the differential squared weighted average for the gradient. This method is beneficial to eliminate the direction of the swing amplitude and to correct the swing amplitude, so the swing amplitude of each dimension is small; on the other hand, the network function converges faster.

The stochastic gradient descent algorithm (SGD) is used as the optimizer of the CNN model, and the SGD updates the network model weights in combination with the gradient and the update weight of the previous iteration. The entire process can be represented by (17) and (18).

$$V_{t+1} = \mu V_t + \alpha \nabla L(W_t), \quad (17)$$

$$W_{t+1} = W_t - V_{t+1}, \quad (18)$$

where W_{t+1} represents the weight of the network after the $t + 1^{th}$ iteration and V_{t+1} is the update amount of the network weight in the $t + 1^{th}$ iteration.

2.3.5. Training and Testing. The WG-CNN framework for bearing fault diagnosis combined with WGAN and CNN is shown in Figure 4. The part marked by the green line represents the training process and the part marked by the blue line represents the testing process. The input to the sample is a three-dimensional mixing matrix consisting of a series of two-dimensional data. The output are the sample classes calculated according to softmax, then the generator loss function of the WGAN model is given by equation (9), the loss function of the discriminator is given by (10), the loss of the CNN model is given by (12), and finally the WGAN model is iteratively optimized by (13) to (16), and the CNN model is optimized by (17) and (18).

We test the model by expanding the number of samples in the dataset, which is useful for a more comprehensive evaluation of the performance of the model, where the input is the original dataset or the enhanced dataset trained by our model, and the output of the model is the corresponding category of the prediction. The classification effect of the model is carried out through related evaluation indicators such as F1-score, precision, and recall.

3. Experiments and Results

The CWRU dataset is a basic dataset for verifying the performance of different machine learning (ML) and deep learning (DL) algorithms. In order to verify the performance of our method in the limited data fault diagnosis, we selected the drive end bearing health and fault data with the motor speed of 1730 rpm and the sampling frequency of 12 k as the original experiment data in the Case Western Reserve University (CWRU) bearing dataset. There are three types of bearing fault locations: ball faults, internal faults, and external faults. Each fault contains three types: 0.007 inches, 0.014 inches, and 0.021 inches, respectively. We have a total of 10 types (0–9, 0 for health and 1–9 for different types and sizes of faults). The specific classification is shown in Table 1.

3.1. Sample Size and Generation Effect. In the first section of the experiments, we evaluated the effect of generating and expanding data by WGAN and solved two challenges first (a total of three challenges) in the limited data fault diagnosis: (1) the bearing fault has serious consequences, especially in the actual production, and the industrial system is not allowed to enter the fault state; (2) most motor faults occur very slowly and follow the degradation path, and the system degradation may take months or even years.

In order to maximize the data generation effect, we conducted a comparative test for the value of the gradient penalty coefficient λ. The experimental results are shown in Table 2. It can be seen from Table 2 that when the gradient penalty coefficient is 10, the experimental results have the highest accuracy. Therefore, in this paper, we set the gradient penalty coefficient to 10 for subsequent experiments.

Firstly, in order to balance the sampling, we selected the same proportion of data from each kind of fault datasets for the experiment. The dataset partition and amount of each dataset are shown in Table 3. A is the original dataset, B is the input dataset of WGAN (randomly choosing 3%, reason is given later), and C is selected as the test dataset. Dataset D is

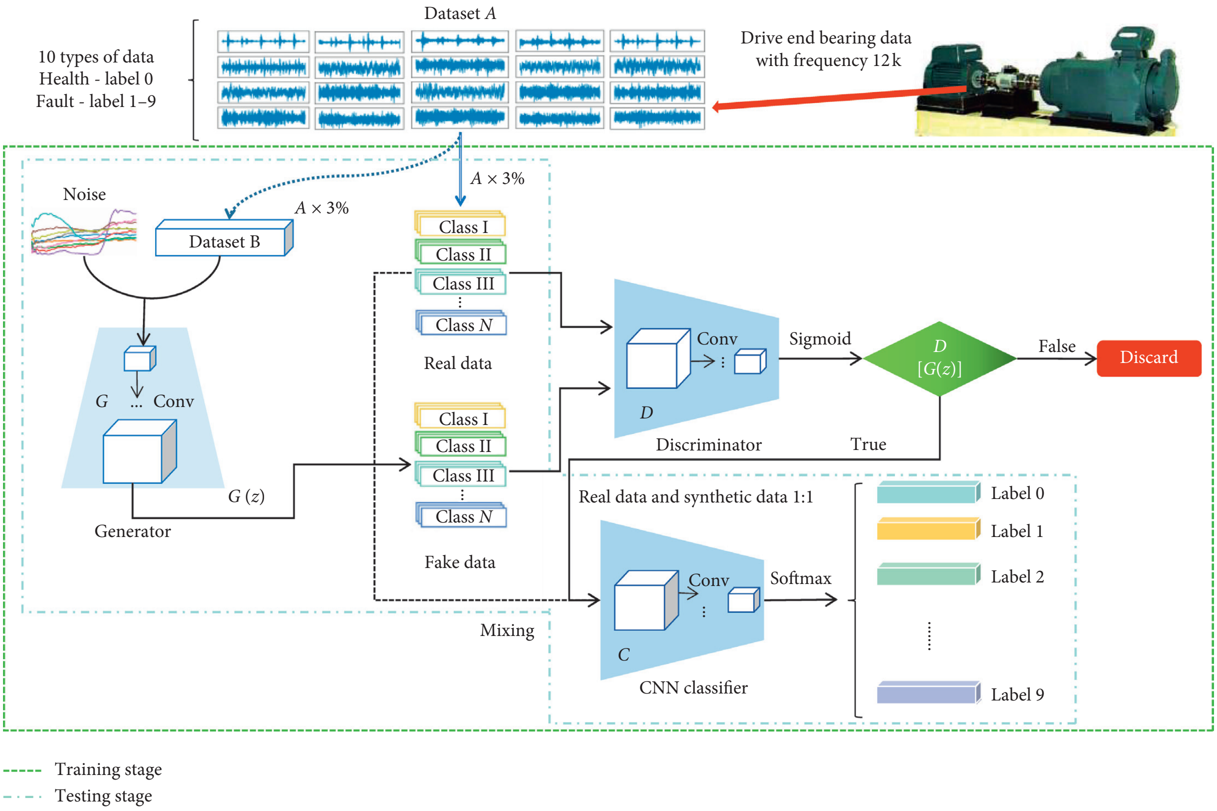


Figure 4: WG-CNN framework. The WG-CNN architecture is mainly composed of three parts: a generator (for generating data), a discriminator (to distinguish between the generated data from G and the data from the original dataset), and a classifier (to evaluate the effect of enhanced data).

Table 1: Type/label/category index.

Type/label/category	Parameter
0	12K-Normal_Baseline
1	12K-DriveEndFault_0.007_Ball
2	12K-DriveEndFault_0.014_Ball
3	12K-DriveEndFault_0.021_Ball
4	12K-DriveEndFault_0.07_Inner
5	12K-DriveEndFault_0.014_Inner
6	12K-DriveEndFault_0.021_Inner
7	12K-DriveEndFault_0.07_Outer
8	12K-DriveEndFault_0.014_Outer
9	12K-DriveEndFault_0.021_Outer

Table 2: Gradient penalty coefficient λ comparison experiment.

λ	1	5	10	15	20
Accuracy	0.94	0.97	0.99	0.94	0.96

the data generated by WGAN, which involves all the ten different types. The last dataset E is the enhanced dataset combined with the original dataset B and the generated dataset D.

Secondly, 14 groups of samples with the numbers of 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1200, 2000, 3000, and 4000 were selected to train the WGAN model. Figures 5 and 6 show the changes of the loss function value of the generator and discriminator with the data amount of 4000. In 100,000 iterations, the WGAN's generator loss value fluctuates from −1.573 to −0.009, and the WGAN's discriminator loss value fluctuates from −0.022 to +2.793. The floating trend changes greatly in the early stage and is stable in the middle and late stages. The values are constantly approaching zero, which shows that WGAN has improved the convergence effect of the traditional GAN significantly.

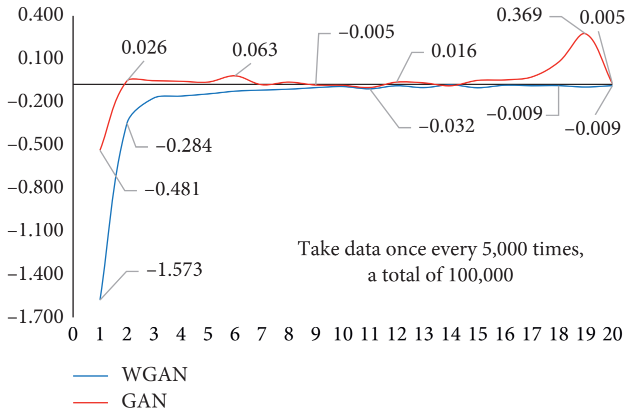
At the same time, in order to quantitatively analyze the generation effects of WGAN and GAN, this paper chooses the Fréchet distance (F) [38] as a metric and quantifies the generation effect by calculating the similarity between the original data and the generated data. The definition of similarity is $S = 1/F$.

The results of the comparative experiment are shown in Table 4. From Table 4, it can be seen that the similarity between the original data and the data generated by WGAN is better, which proves the correctness of WGAN in this paper.

Thirdly, we use dataset B as the training set and dataset C as the test set of the proposed model. In order to design the optimal CNN classifier structure, we conducted a comparative experiment on the number of convolution kernels and the type of activation function. The results of the experiment are shown in Table 5. At the same time, the effect of batch processing size and learning rate on the model test accuracy was analyzed experimentally. The results are shown in Figure 7.

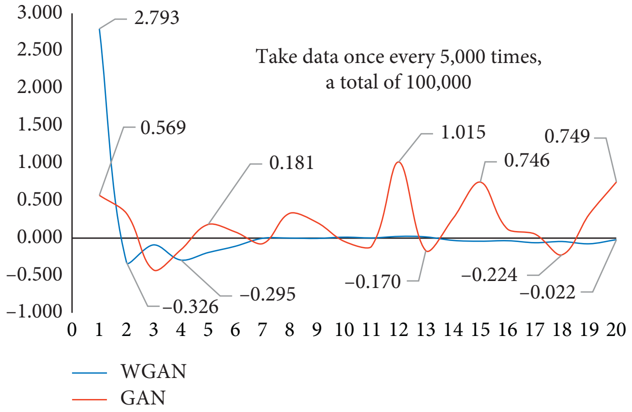
Table 3: Dataset partition and amount.

Type	A Original dataset	B ($A \times 3\%$) Input	C ($A \times 20\%$) Test	D (WGAN) Output	E ($B + D$) Enhancement
0	485,643	14,569	97,128	14,569	29,138
1	121,556	3,647	24,308	3,647	7,294
2	122,136	3,664	24,424	3,664	7,328
3	122,136	3,664	24,424	3,664	7,328
4	122,917	3,688	24,580	3,688	7,376
5	121,701	3,651	24,340	3,651	7,302
6	121,991	3,660	24,396	3,660	7,320
7	122,571	3,677	24,512	3,677	7,354
8	121,991	3,660	24,396	3,660	7,320
9	121,991	3,660	24,396	3,660	7,320



Take data once every 5,000 times, a total of 100,000

Figure 5: The generator loss value.



Take data once every 5,000 times, a total of 100,000

Figure 6: The discriminator loss value.

The specific structure of optimized CNN is shown in Table 6. A total of 14 groups, which involve the numbers of 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1200, 2000, 3000, and 4000, respectively, were trained and tested. Figure 8 shows the partial confusion matrix of the test results. Among them, health type 0 and fault types 4 and 6 are easy to identify, but fault types 3 and 9 are not easy to identify. It should be noted that as the amount of training data increases, the test results become better. When the amount of data in the training set reaches 4000, the method shows satisfactory classification result, and the test accuracy is more than 98%. Therefore, in the next experiments, we usually choose 3% (4000/121556≈0.03) as the minimum selection proportion form original data. Experimental results show that the WGAN model can reduce the amount of data for training significantly; at the same time, it can meet the requirements of accuracy.

Finally, we selected representative categories 2, 4, 7, and 8 to display the effect of sample data generation. As shown in Figure 9, "real" is the real sample and "synthetic" is the generated sample. The orange line in the middle represents the mean of the data, the green line above is the mean plus the variance, and the blue line below is the mean minus the variance. Since the generated samples are standardized, they have a little different from the actual data ordinate values.

Among the four types of faults shown, categories 7 and 8 produced the best results, and the trend of generated sample was basically the same as the real sample. The variance of categories 7 and 8 is significantly larger than categories 2 and 4. It shows that the generated data are inevitably affected by the noise in real environments, and the applicability of the generated data needs to be further improved. In the following section, we will explore the comparison between different training data amount and model accuracy of the deep learning algorithm based on CNN classification.

3.2. *Enhancement Data and Accuracy.*

When selecting training samples in the CWRU dataset, many previous papers cannot guarantee balanced sampling, which means that the proportion of data samples selected from the normal state and the fault state is not close to 1 : 1. If most training sets are from health data, the learned features will not be suitable for fault classification, so this paper proposes an average sampling method in each type to deal with data imbalance.

On this basis, in order to describe the effect of different training sets more accurately, the average accuracy should not be used as the only index to evaluate the algorithm. Other indexes should be used to measure the effectiveness and reliability of the algorithm, such as precision, recall rate, specificity, and $F1$-score.

Precision is the ratio of correctly classified positive samples to the number of all classified positive samples. Recall (in binary classification also known as sensitivity) is

Table 4: Fréchet distance and similarity of original and generated data.

		0	1	2	3	4	5	6	7	8	9
GAN	F	1.31	2.79	1.48	2.47	1.48	2.59	1.68	2.65	2.16	2.46
	S	0.76	0.36	0.68	0.40	0.68	0.39	0.60	0.38	0.46	0.41
WGAN	F	0.64	1.25	0.99	1.38	1.19	1.21	0.67	1.64	1.32	1.21
	S	1.56	0.8	1.01	0.72	0.84	0.83	1.49	0.61	0.76	0.83

Table 5: Comparison of contrast kernel settings.

Experimental group	Number of convolution kernels in different layers			Activation function	Test accuracy
	Layer 1	Layer 2	Layer 3		
1	8	16	32	*ReLU*	0.96
2	16	32	64	*ReLU*	0.99
3	32	64	128	*ReLU*	0.98
4	8	16	32	*Leaky ReLU*	0.95
5	16	32	64	*Leaky ReLU*	1.00
6	32	64	128	*Leaky ReLU*	0.98
7	8	16	32	*Tanh*	0.96
8	16	32	64	*Tanh*	0.99
9	32	64	128	*Tanh*	0.97

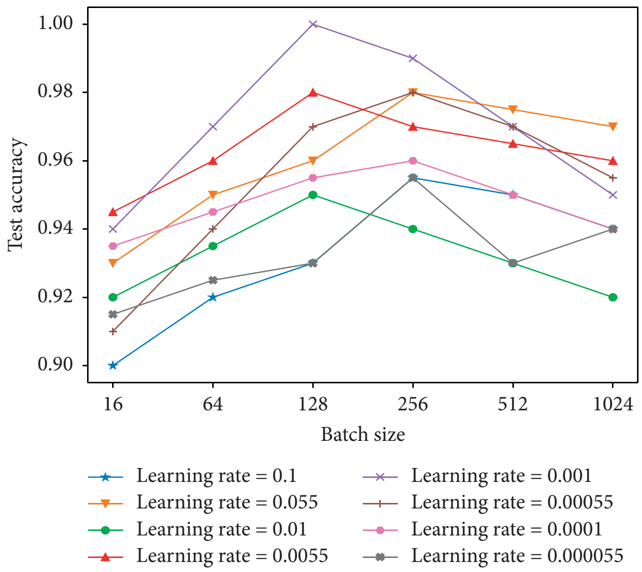


Figure 7: Comparison of learning rate size and batch size experiment.

the ratio of correctly classified positive samples to the number of all of the positive annotated samples. $F1$-score is the harmonic mean of precision and recall. Accuracy is simply the ratio of correctly classified samples to the number of samples.

TP (true positive)—samples annotated and classified by the positive label.

TN (true negative)—samples annotated and classified by the negative label.

FP (false positive)—samples annotated as negative but classified as positive.

FN (false negative)—samples annotated as positive but classified as negative.

$$\begin{aligned}
\text{Precision}\,(\%) &= \left(\frac{\text{TP}}{\text{TP}+\text{FP}}\right) \times 100\%,\\
\text{recall}\,(\%) = \text{sensitivity}\,(\%) &= \left(\frac{\text{TP}}{\text{TP}+\text{FN}}\right) \times 100\%,\\
\text{specificity}\,(\%) &= \left(\frac{\text{TN}}{\text{FP}+\text{TN}}\right) \times 100\%,\\
F1\,(\%) &= \left(2 \times \frac{\text{prec} \times \text{rec}}{\text{prec} + \text{rec}}\right) \times 100\%,\\
\text{accuracy}\,(\%) &= \left(\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}\right) \times 100\%.
\end{aligned} \tag{19}$$

In order to compare the differences between the generated dataset and the original dataset, we conducted the following experiments with the control variable method. In the first section, we concluded through experiments that when the amount of data in the set reaches 4000, the WGAN model has performed well, and the training dataset sample is only 3% of the original data. Therefore, in the experiments of this section, we randomly choose 3% of the original data as dataset B to train WGAN to generate a new dataset.

Firstly, CNN was trained on dataset B, and dataset C was used as the test set to verify the fault classification effect of the model. The results are shown in Table 7 and Figure 10(a).

Secondly, we trained on dataset D and tested on dataset C to verify the fault classification effect of the generated dataset. The results are shown in Table 8 and Figure 10(b). As can be seen from the comparison experiment in Figures 10(a) and 10(b), the precision, recall rate, and $F1$-score of the generated data are very similar to the original

Table 6: CNN structures and parameters used in this experiment.

No.	Network layer	Kernel number	Kernel size/strides	Output size
1	Convolution1	16	3/1	2998 × 16
2	Convolution1	16	3/1	2996 × 16
3	Convolution1	16	3/1	2994 × 16
4	Maxpooling1	16	2/1	1497 × 16
5	Convolution2	32	3/1	1495 × 32
6	Convolution2	32	3/1	1493 × 32
7	Convolution2	32	3/1	1491 × 32
8	Maxpooling2	32	2/1	745 × 32
9	Convolution3	64	3/1	743 × 64
10	Convolution3	64	3/1	741 × 64
11	Convolution3	64	3/1	739 × 64
12	Maxpooling3	64	2/1	369 × 64
13		Flatten		23616
14		Dense1 + Dropout1		1024
15		Dense2 + Dropout2		512
16		Dense3 + Dropout3		10

dataset, which shows that CNN still has better performance in the generated data and the generated data are available.

Thirdly, the experiment analyzes the classification effect of enhanced data. We input the original data into WGAN, training the model through the confrontation between the generator and the discriminator, and generate high-quality dataset *D*. Then, the original dataset *B* and the generated dataset *D* are combined to get the enhanced dataset *E*; we train on dataset *E* and test on dataset *C*. The test results are shown in Table 9 and Figure10(c).

It can be seen from Figures 10(a)–10(c) that compared with the original dataset *B* and the generated dataset *D*, the CNN classification effect under the enhanced dataset *E* is greatly improved, and the *F*1-score of four items reaches 100%. Figure 10(d) shows the average value of each indicator of sklearn evaluation function of the three comparative experiments. The accuracy of CNN classification based on enhanced dataset *E* has obvious advantages. The experiment shows that WGAN is an effective data enhancement strategy for bearing fault classification.

To compare the performance of different enhancement algorithms, we randomly select 20% samples as the training set and 40% samples as the test set from original dataset of 5 random types of faults. For the other 5 types of faults, 10% of the samples are randomly selected as the training set and 40% of the samples are used as the test set. For the rare data, in addition to the SVM method, the other comparison methods adopt different enhancement strategies so that the number of training sets for each type of faults is 20% of the original data. SVM [39, 40], CNN with oversampling, CNN with downsampling, and GAN-CNN are compared with the proposed model (WG-CNN). The experiment results are shown in Table 10.

It can be seen from Table 10 that the benchmark model (SVM) has a poor classification effect on unbalanced datasets, and the average accuracy is only 75%. Two general data enhancement methods are used to enhance the CNN classifier, respectively. The accuracy of the classifier enhanced by the downsampling method is higher than that of the oversampling enhanced classifier. We also compare the original GAN generated data with the proposed model. The results show that the model proposed in this paper has a higher result, which can learn the distribution and characteristics of the data more accurately and provide high-quality generated data for the classification task.

3.3. Comparison of Different Algorithms. In this section, we test different algorithms to compare the performance. For 10 different types of datasets, 20% of the original data were randomly selected as the input of the different algorithms. At the same time, 40% of the original data were randomly selected as the test dataset. In order to select the number of training iterations in the experiment, we define the algorithm efficiency (*AE*):

$$AE = \frac{\text{test} \quad \text{accuracy}}{\text{total} \quad \text{time}} * 100. \tag{20}$$

Through 5 sets of comparative experiments, the efficiency of the algorithm under different iterations is obtained, as shown in Table 11.

Through the comparison experiment of algorithm efficiency, we found that when the number of iterations is 10000, the algorithm efficiency is up to 1.519, but the accuracy of the algorithm test at this time is only 0.41, so 10000 cannot be used as the number of iterations of the experiment in this paper. Combining algorithm efficiency and algorithm test accuracy, we found that when the number of iterations is 100000, the efficiency and accuracy of the algorithm have reached the desired effect, so 100000 is used as the number of iterations in the subsequent comparative experiments. At the same time, this also verifies the correctness of the selection of the number of iterations in our previous experiments.

Then, the experimental analysis for each comparative algorithm was performed. References give the proposed paper of the model, and we use each proposed model to conduct experimental analysis under the unified dataset. Table 12 shows the classification accuracy of different DL algorithms on the CWRU datasets.

It can be easily observed that the test accuracy of all CNN-based deep learning algorithms exceeds 70%, which proves the feasibility and effectiveness of using CNN for bearing fault diagnosis. However, under the same training data and test data settings, the accuracy between different models is very different. Among them, the original CNN model has the lowest accuracy for fault classification, which reached only 72.4%. The model with the highest classification accuracy for the dataset is the model proposed in this paper (WG-CAN), which reached 100%. This proves the effectiveness and practicability of the proposed algorithm.

As can be seen from the data in Table 12, the proposed WG-CNN learning algorithm can take into account the accuracy of classification while significantly reducing its dependence on the original data.

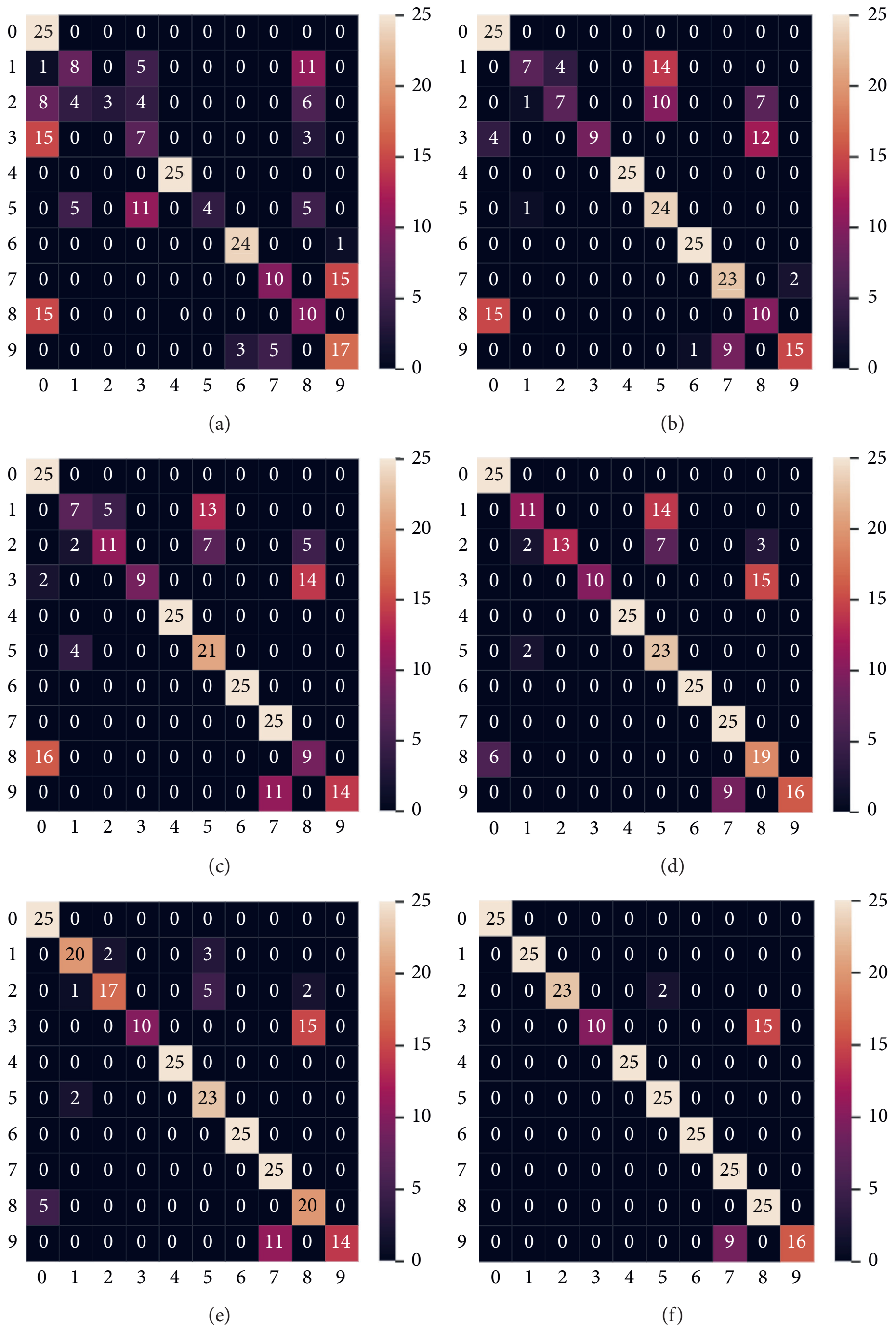
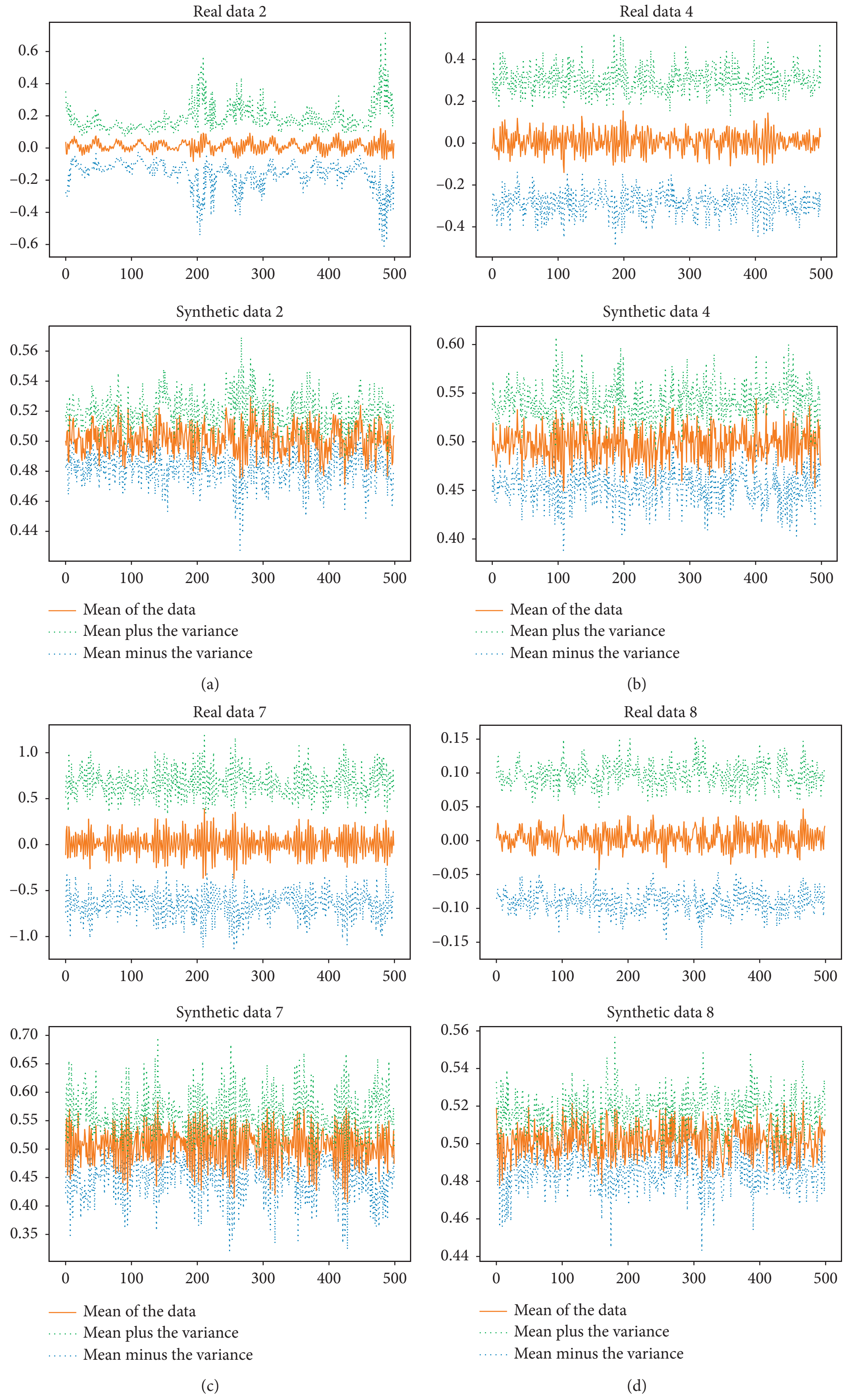


FIGURE 8: The confusion matrix for different training sample results: (a) 100 training samples; (b) 500 training samples; (c) 700 training samples; (d) 1200 training samples; (e) 2000 training samples; (f) 4000 training samples.

3.4. Noise and Classification Results. The fourth section of the experiment is a test of the stability and robustness of the WG-CNN algorithm. In fact, all bearing defects in the CWRU dataset are drilled or engraved manually, which is easier to identify than actual bearing wear and general roughness due to aging. It can be seen from Table 12 that even with the classic and ordinary CNN, the CWRU dataset can achieve 70% excellent classification accuracy. This shows that the dataset contains relatively simple features, which can be easily extracted by a variety of DL methods. Therefore, in the experiment, the original data provided by CWRU were used to train the model, and then the white Gaussian test samples of different SNR were added for testing. The SNR varies from −4 dB to 8 dB. Dataset *B* is used as the training set of CNN model and the proposed model, and different test sets are formed by adding different proportions of noise to dataset *C*.

In this experiment, we evaluate the classification effect of WG-CNN algorithm proposed in this paper to determine



(a)

(b)

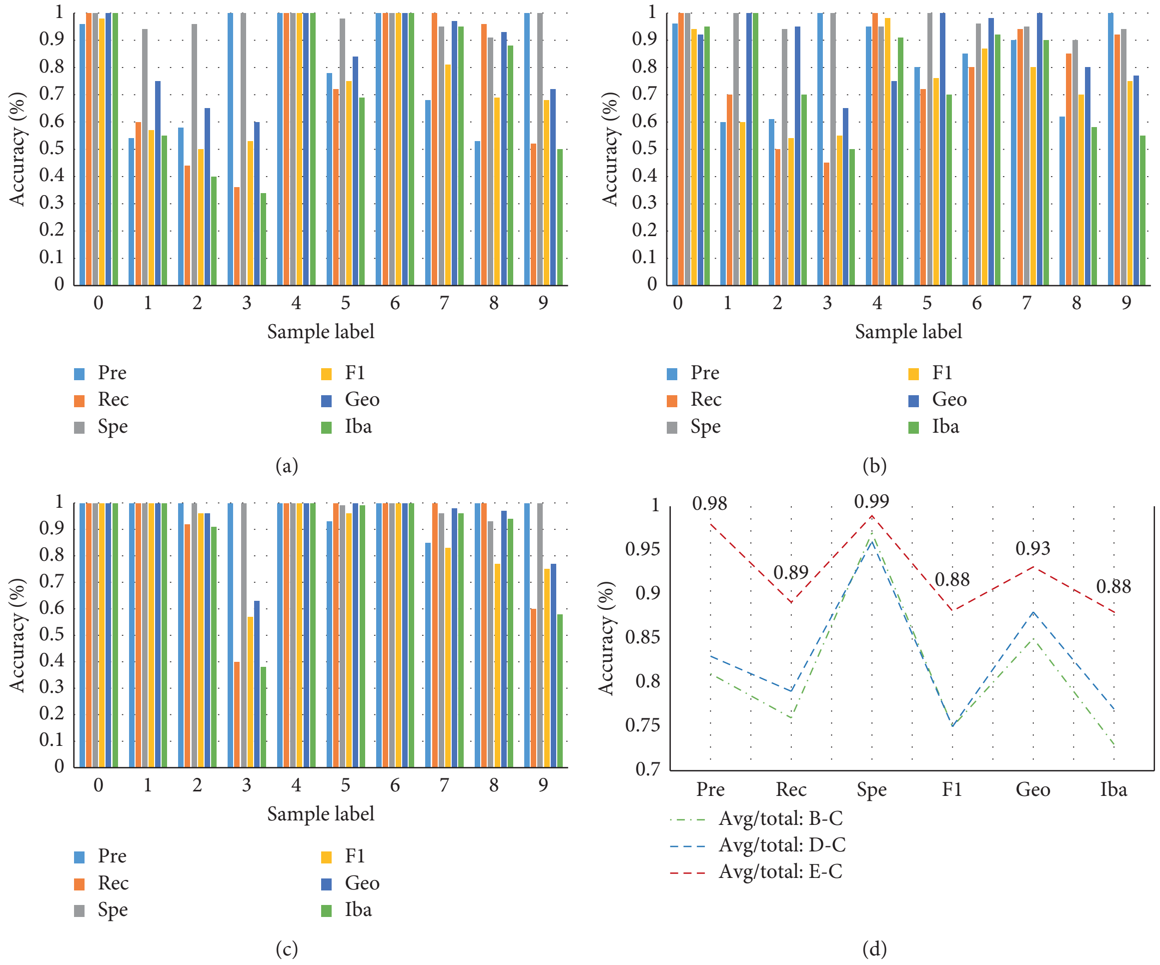
(c)

(d)

Figure 9: Real and synthetic data effect comparison display for four types: (a) type-2; (b) type-4; (c) type-7; (d) type-8.

Table 7: B-C dataset (sklearn evaluation details).

Type	Pre	Rec	Spe	*F1*	Geo	Iba
0	0.96	1.00	1.00	0.98	1.00	1.00
1	0.54	0.60	0.94	0.57	0.75	0.55
2	0.58	0.44	0.96	0.50	0.65	0.40
3	1.00	0.36	1.00	0.53	0.60	0.34
4	1.00	1.00	1.00	1.00	1.00	1.00
5	0.78	0.72	0.98	0.75	0.84	0.69
6	1.00	1.00	1.00	1.00	1.00	1.00
7	0.68	1.00	0.95	0.81	0.97	0.95
8	0.53	0.96	0.91	0.69	0.93	0.88
9	1.00	0.52	1.00	0.68	0.72	0.50
Avg/total	0.81	0.76	0.97	0.75	0.85	0.73



(a)

(b)

(c)

(d)

Figure 10: Sklearn evaluation function results: (a) B-C dataset; (b) D-C dataset; (c) E-C dataset; (d) comparison of experimental averages.

Table 8: D-C dataset (sklearn evaluation details).

Type	Pre	Rec	Spe	*F1*	Geo	Iba
0	0.96	1.00	1.00	0.94	0.92	0.95
1	0.60	0.70	1.00	0.60	1.00	1.00
2	0.61	0.50	0.94	0.54	0.95	0.70
3	1.00	0.45	1.00	0.55	0.65	0.50
4	0.95	1.00	0.95	0.98	0.75	0.91
5	0.80	0.72	1.00	0.76	1.00	0.70
6	0.85	0.80	0.96	0.87	0.98	0.92
7	0.90	0.94	0.95	0.80	1.00	0.90
8	0.62	0.85	0.90	0.70	0.80	0.58
9	1.00	0.92	0.94	0.75	0.77	0.55
Avg/total	0.83	0.79	0.96	0.75	0.88	0.77

Table 9: E-C dataset (sklearn evaluation details).

Type	Pre	Rec	Spe	*F*l	Geo	Iba
0	1.00	1.00	1.00	1.00	1.00	1.00
1	1.00	1.00	1.00	1.00	1.00	1.00
2	1.00	0.92	1.00	0.96	0.96	0.91
3	1.00	0.40	1.00	0.57	0.63	0.38
4	1.00	1.00	1.00	1.00	1.00	1.00
5	0.93	1.00	0.99	0.96	1.00	0.99
6	1.00	1.00	1.00	1.00	1.00	1.00
7	0.85	1.00	0.96	0.83	0.98	0.96
8	1.00	1.00	0.93	0.77	0.97	0.94
9	1.00	0.60	1.00	0.75	0.77	0.58
Avg/total	0.98	0.89	0.99	0.88	0.93	0.88

Table 10: Comparison between different enhanced algorithms.

Enhanced algorithms	Pre	Recall	*F*1	Acc
SVM	0.77	0.7	0.73	0.75
CNN with oversampling	0.83	0.78	0.8	0.8
CNN with downsampling	0.82	0.77	0.79	0.81
GAN-CNN	0.91	0.9	0.9	0.91
WG-CNN	**0.96**	**0.99**	**0.97**	**0.99**

Table 11: Comparison of algorithm efficiency of the proposed model.

Number of iterations	Training time (min)	Testing time (min)	Total time (min)	Test accuracy	Algorithm efficiency
10000	25	2	27	0.41	1.519
50000	64	2	66	0.54	0.818
100000	118	2	120	0.99	0.825
150000	156	2	158	0.98	0.620
200000	204	2	206	0.99	0.481

Table 12: Performance comparison of DL algorithms based on CWRU dataset.

Core algorithm	Hidden layers	Classifier	Average accuracy	Reference
CNN	4	*Softmax*	72.40%	[41]
CNN with training interface	12	*Softmax*	75.51%	[42]
Adaptive CNN	3	*Softmax*	87.94%	[43]
Multiscale deep CNN	9	*Softmax*	78.67%	[44]
IDS-CNN	3	*Softmax*	78.32%	[45]
PSPP-CNN	9	*Softmax*	97.19%	[28]
CNN	4	*Softmax*	89.98%	[46]
CNN-LSTM	3	*Softmax*	89.67%	[47]
CNN-based LiftingNet	6	*FC layer*	89.61%	[27]
CNN based on LeNet-5	8	*FC layer*	89.97%	[26]
SN-SSGAN	18	*Softmax*	95.78%	[35]
K-means WGAN-GP	12	*RVM/SVM*	97.65%	[48]
WG-CNN	***9***	***Softmax***	***100%***	/

Table 13: Signal-to-noise ratio.

SNR	$P_{\text{signal}}:P_{\text{noise}}$
−4	0.5 : 1
−2	0.8 : 1
0	1 : 1
2	1.6 : 1
4	2.5 : 1
6	4 : 1
8	6.3 : 1

whether we can solve the third challenge in the limited data fault diagnosis: the complex working conditions of mechanical bearings, so it is difficult to collect and mark enough training samples with reasonable noise. In the following formula and Table 13, signal-to-noise ratio (SNR) is defined as the ratio of signal power to noise power, usually expressed in decibels:

$$SNR_{dB} = 10\log_{10}\left(P_{\text{signal}}/P_{\text{noise}}\right), \tag{21}$$

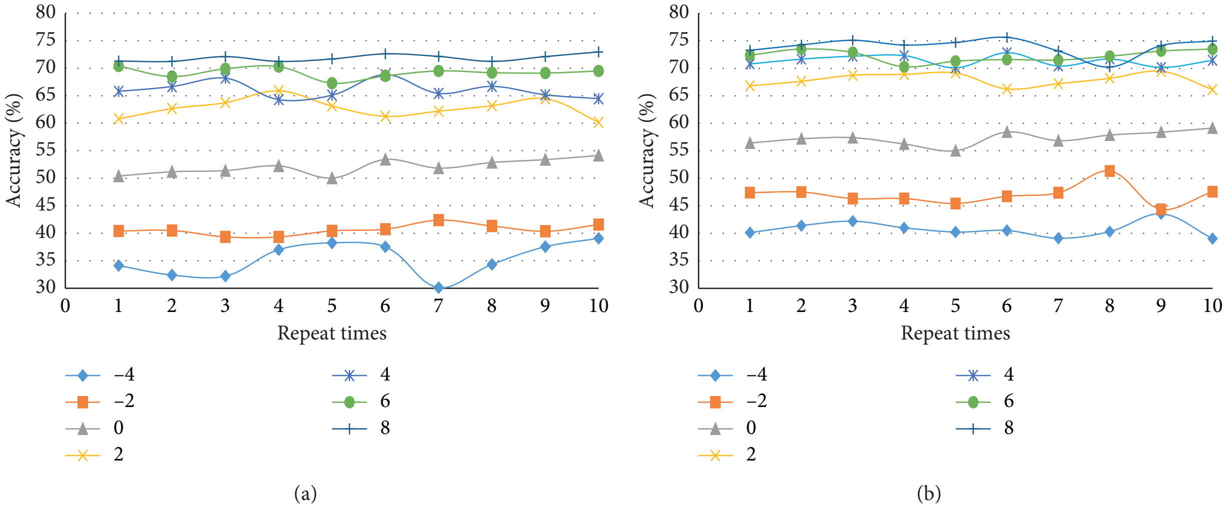


Figure 11: Accuracy under different noise: (a) CNN; (b) WG-CNN.

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

Figure 11 shows the comparison of the test results of the WG-CNN model in different noise environments. In the case of 4 dB and 6 dB SNR, the difference between the training set and the test set was small, and the test results were also better. When the SNR is −4 dB and −2 dB, the difference between the training set and the test set is large, and the accuracy is lower. WG-CNN is more robust and stable than traditional CNN.

Comparing Figures 11(a) and 11(b), we find that the average accuracy of WG-CNN is 40% at −4 dB and that of CNN is 32% at −4 dB. At high SNR (4 dB, 6 dB, and 8 dB), the average accuracy of WG-CNN is slightly higher than that of CNN, and the curve coincidence is significantly higher than that of CNN. At this time, the noise power is less than the signal power, which is closer to the actual working environment. Experiments show that WG-CNN has stronger robustness and stability than traditional CNN in noisy environments.

4. Conclusions

This paper proposes a few-shot learning neural network method for bearing fault diagnosis with limited data. This method is based on the WG-CNN model which combines Wasserstein generative adversarial network (WGAN) and convolutional neural network (CNN). It can be used for deep learning on limited samples, and it can effectively improve the performance of fault diagnosis. We solved three major challenges in the field of bearing fault diagnosis. Experiments show that WG-CNN can significantly reduce the number of training samples. When only 20% of the standard Case Western Reserve University (CWRU) bearing fault diagnosis reference dataset is selected, the classification accuracy reaches 100%, which is the highest among all the comparison papers. In the future, we plan to combine WG-CNN with transfer learning and domain adaptation and use this method to diagnose bearing fault in the actual field.

Data Availability

The simulation data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This study was supported in part by the Aeronautical Science Foundation of China under grant no. 2015ZB54007, the Scientific Research Foundation of Liaoning Education Department under grant nos. L201627, L201750, and L201704, and the Youth Program of National Natural Science Foundation of China under grant no. 51905355.

References

[1] F. Filippetti, G. Franceschini, C. Tassoni, and P. Vas, “Recent developments of induction motor drives fault diagnosis using AI techniques,” *IEEE Transactions on Industrial Electronics*, vol. 47, no. 5, pp. 994–1004, 2000.

[2] M. A. Awadallah and M. M. Morcos, “Application of AI tools in fault diagnosis of electrical machines and drives-an overview,” *IEEE Transactions on Energy Conversion*, vol. 18, no. 2, pp. 245–251, 2003.

[3] R. R. Schoen, B. K. Lin, T. G. Habetler, J. H. Schlag, and S. Farag, “An unsupervised, on-line system for induction motor fault detection using stator current monitoring,” *IEEE Transactions on Industry Applications*, vol. 31, no. 6, pp. 1280–1286, 1995.

[4] B. Li, M.-Y. Chow, Y. Tipsuwan, and J. C. Hung, “Neural-network-based motor rolling bearing fault diagnosis,” *IEEE Transactions on Industrial Electronics*, vol. 47, no. 5, pp. 1060–1069, 2000.

[5] M. Cococcioni, B. Lazzerini, and S. L. Volpi, “Robust diagnosis of rolling element bearings based on classification

techniques," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2256–2263, 2013.

[6] S. Qian, X. Yang, J. Huang, and H. Zhang, "Application of new training method combined with feedforward artificial neural network for rolling bearing fault diagnosis," in *Proceedings of the International Conference of Mechatronics Machine Vision in Practice (M2VIP)*, pp. 1–6, Nanjing, China, November 2016.

[7] S. Lu and X. Wang, "PCA-based feature selection scheme for machine defect classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 53, no. 6, pp. 1517–1525, 2004.

[8] J. Harmouche, C. Delpha, and D. Diallo, "A global approach for the classification of bearing faults conditions using spectral features," in *Proceedings of the 39th Annual Conference of the IEEE Industrial Electronics Society*, pp. 7352–7357, Vienna, Austria, November 2013.

[9] X. Xue and J. Zhou, "A hybrid fault diagnosis approach based on mixed-domain state features for rotating machinery," *ISA Transactions*, vol. 66, pp. 284–295, 2017.

[10] A. Rojas and A. K. Nandi, "Detection and classification of rollingelement bearing faults using support vector machines," in *Proceedings of the IEEE Workshop on Machine Learning for Signal Processing*, pp. 153–158, Mystic, CT, USA, September 2005.

[11] M. M. Ettefagh, M. Ghaemi, and M. Yazdanian Asr, "Bearing fault diagnosis using hybrid genetic algorithm K-means clustering," in *Proceedings of the IEEE International Symposium on Innovations in Intelligent Systems and Applications*, pp. 84–89, Alberobello, Italy, June 2014.

[12] J. Tian, C. Morillo, M. H. Azarian, and M. Pecht, "Motor bearing fault detection using spectral Kurtosis-based feature extraction coupled with K-nearest neighbor distance analysis," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1793–1803, 2016.

[13] T. W. Rauber, F. de Assis Boldt, and F. M. Varejao, "Heterogeneous feature models and feature selection applied to bearing fault diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 637–646, 2015.

[14] J. Wu, C. Wu, S. Cao, S. W. Or, C. Deng, and X. Shao, "Degradation data-driven time-to-failure prognostics approach for rolling element bearings in electrical machines," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 1, pp. 529–539, 2019.

[15] Q. Hu, A. Qin, Q. Zhang, J. He, and G. Sun, "Fault diagnosis based on weighted extreme learning machine with wavelet packet decomposition and KPCA," *IEEE Sensors Journal*, vol. 18, no. 20, pp. 8472–8483, 2018.

[16] R. Zhang, H. Tao, L. Wu, and Y. Guan, "Transfer learning with neural networks for bearing fault diagnosis in changing working conditions," *IEEE Access*, vol. 5, pp. 14347–14357, 2017.

[17] Z. Wang, Q. Zhang, J. Xiong, M. Xiao, G. Sun, and J. He, "Fault diagnosis of a rolling bearing using wavelet packet denoising and random forests," *IEEE Sensors Journal*, vol. 17, no. 17, pp. 5581–5588, 2017.

[18] T. Yang, H. Pen, Z. Wang, and C. S. Chang, "Feature knowledge based fault detection of induction motors through the analysis of stator current data," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 549–558, 2016.

[19] B. Yao, P. Zhen, L. Wu, and Y. Guan, "Rolling element bearing fault diagnosis using improved manifold learning," *IEEE Access*, vol. 5, pp. 6027–6035, 2017.

[20] B. Wang, H. Pan, and W. Yang, "A complementary approach for fault diagnosis of rolling bearing using canonical variate analysis based short-time energy feature," *Journal of Vibration and Control*, vol. 24, no. 18, pp. 4195–4210, 2018.

[21] R. Razavi-Far, M. Farajzadeh-Zanjani, and M. Saif, "An integrated class-imbalanced learning scheme for diagnosing bearing defects in induction motors," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 2758–2769, 2017.

[22] J. Tian, M. H. Azarian, M. Pecht, G. Niu, and C. Li, "An ensemble learning-based fault diagnosis method for rotating machinery," in *Proceedings of the 2017 Prognostics and System Health Management Conference*, pp. 1–6, Harbin, China, October 2017.

[23] Q. Fu, B. Jing, P. He, S. Si, and Y. Wang, "Fault feature selection and diagnosis of rolling bearings based on EEMD and optimized Elman_ AdaBoost algorithm," *IEEE Sensors Journal*, vol. 18, no. 12, pp. 5024–5034, 2018.

[24] Y. Lu, R. Xie, and S.-Y. Liang, "Adaptive online dictionary learning for bearing fault diagnosis," *The International Journal of Advanced Manufacturing Technology*, vol. 101, pp. 1–8, 2019.

[25] S. Singh, C. Q. Howard, and C. Hansen, "Convolutional neural network based fault detection for rotating machinery," *Journal of Sound and Vibration*, vol. 377, pp. 331–345, 2016.

[26] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5990–5998, 2018.

[27] J. Pan, Y. Zi, J. Chen, Z. Zhou, and B. Wang, "LiftingNet: a novel deep learning network with layerwise feature learning from noisy mechanical data for fault classification," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 6, pp. 4973–4982, 2018.

[28] S. Guo, T. Yang, W. Gao, C. Zhang, and Y. Zhang, "An intelligent fault diagnosis method for bearings with variable rotating speed based on Pythagorean spatial pyramid pooling CNN," *Sensors*, vol. 18, 2018.

[29] W. Qian, S. Li, J. Wang, Z. An, and X. Jiang, "An intelligent fault diagnosis framework for raw vibration signals: adaptive overlapping convolutional neural network," *Measurement Science and Technology*, vol. 29, no. 6, 2018.

[30] Y. Chen, G. Peng, C. Xie et al., "ACDIN: bridging the gap between artificial and real bearing damages for bearing fault diagnosis," *Neurocomputing*, vol. 294, pp. 61–71, 2018.

[31] Z. Chen and W. Li, "Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 7, pp. 1693–1702, 2017.

[32] W. Abed, S. Sharma, R. Sutton, and A. Motwani, "A robust bearing fault detection and diagnosis technique for brushless DC motors under non-stationary operating conditions," *Journal of Control, Automation and Electrical Systems*, vol. 26, no. 3, pp. 241–254, 2015.

[33] H. Shao, H. Jiang, Y. Lin, and X. Li, "A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders," *Mechanical Systems and Signal Processing*, vol. 102, pp. 278–297, 2018.

[34] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the Annual Conference on Neural Information Processing Systems*, Montreal, Canada, December 2014.

[35] D. Zhao, F. Liu, and M. He, "bearing fault diagnosis based on the switchable normalization SSGAN with 1-D representation

of vibration signals as input," *Sensors*, vol. 19, no. 9, p. 2000, 2019.

[36] M. Arjovskyetal, "Wasserstein generative adversaria networks," in *Proceedings of the International Conference on Machine Learning*, Sydney, Australia, August 2017.

[37] I. Gulrajani, "Improved training of wasserstein GANs," in *Proceedings of the Advances in Neural Information Processing Systems*, Long Beach, CA, USA, December 2017.

[38] T. Eiter and H. Mannila, "Computing discrete Fréchet distance," Tech. Report CD-TR 94/64, Information Systems Department, Technical University of Vienna, Vienna, Austria, 1994.

[39] J. Zuo, J. Chen, Y. Tan, M. Wang, and L. Wen, "A multi-agent collaborative work planning strategy based on AFSA-PSO algorithm," in *Proceedins of the 2019 International Conference on Robots & Intelligent System (ICRIS)*, pp. 254–257, Haikou, China, June 2019.

[40] N. Qu, J. Zuo, J. Chen, and Z. Li, "Series arc fault detection of indoor power distribution system based on LVQ-NN and PSO-SVM," *IEEE Access*, vol. 7, pp. 184020–184028, 2019.

[41] C. Lu, Z. Wang, and B. Zhou, "Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification," *Advanced Engineering Informatics*, vol. 32, pp. 139–151, 2017.

[42] W. Zhang, C. Li, G. Peng, Y. Chen, and Z. Zhang, "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.

[43] X. Guo, L. Chen, and C. Shen, "Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis," *Measurement*, vol. 93, pp. 490–502, 2016.

[44] Z. Zhuang and Q. Wei, "Intelligent fault diagnosis of rolling bearing using one-dimensional multi-scale deep convolutional neural network based health state classification," in *Proceedings of the 2018 IEEE 15th International Conference on Networking Sensing and Control*, pp. 1–6, Zhuhai, China, March 2018.

[45] S. Li, G. Liu, X. Tang, J. Lu, and J. Hu, "An ensemble deep convolutional neural network model with improved D-S evidence fusion for bearing fault diagnosis," *Sensors*, vol. 17, 2017.

[46] M. Xia, T. Li, L. Xu, L. Liu, and C. W. de Silva, "Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 101–110, 2018.

[47] H. Pan, X. He, S. Tang, and F. Meng, "An improved bearing fault diagnosis method using one-dimensional CNN and LSTM," *Journal of Mechanical Engineering*, vol. 64, no. 7-8, pp. 443–452, 2018.

[48] Z. Zhao, R. Zhou, and Z. Dong, "Aero-engine faults diagnosis based on K-means improved wasserstein GAN and relevant vector machine," in *Proceedings of the 2019 Chinese Control Conference (CCC)*, Guangzhou, China, July 2019.