

Research Article

Bearing Remaining Useful Life Prediction Based on AdCNN and CWGAN under Few Samples

Junfeng Man,^{1,2} Minglei Zheng ,² Yi Liu ,³ Yiping Shen,⁴ and Qianqian Li ²

¹School of Computer, Hunan First Normal University, Changsha 410205, China

²School of Computer, Hunan University of Technology, Zhuzhou 412007, China

³National Advanced Rail Transit Equipment Innovation Center, Zhuzhou 412007, China

⁴Hunan Provincial Key Laboratory of Health Maintenance for Mechanical Equipment, Hunan University of Science and Technology, Xiangtan 411201, China

Correspondence should be addressed to Qianqian Li; 25716332@qq.com

Received 8 December 2021; Revised 2 March 2022; Accepted 18 March 2022; Published 30 June 2022

Academic Editor: Yi Qin

Copyright © 2022 Junfeng Man et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

At present, deep learning is widely used to predict the remaining useful life (RUL) of rotation machinery in failure prediction and health management (PHM). However, in the actual manufacturing process, massive rotating machinery data are not easily obtained, which will lead to the decline of the prediction accuracy of the data-driven deep learning method. Firstly, a novel prognostic framework is proposed, which is comprised of conditional Wasserstein distance-based generative adversarial networks (CWGAN) and adversarial convolution neural networks (AdCNN), which can stably generate high-quality training samples to augment the bearing degradation dataset and solve the problem of few samples. Then, the bearing RUL prediction method is realized by inputting the monitoring data into the one-dimensional convolutional neural network (1DCNN) for adversarial training. Via the bearing degradation dataset of the IEEE 2012 PHM data challenge, the reliability of the proposed method is verified. Finally, experimental results show that our approach is better than others in RUL prediction on average absolute deviation and average square root error.

1. Introduction

Bearings play a particularly vital role in modern industry. The prediction of bearings life can greatly reduce maintenance costs and optimize resource allocation, which can be very helpful to improve equipment reliability [1]. In recent years, with the rapid development of sensor technology and computer technology, the data collected by industrial machinery monitoring is more and more abundant, which has great potential value, and is vital for equipment health analysis and the prediction of remaining useful life [2, 3].

Bearing remaining useful life prediction mainly includes data acquisition, feature extraction and selection, and model establishment [4, 5]. Data acquisition mainly collects monitoring information of bearing operation through sensors, such as vibrational signals and acoustical emission signals. It is one of the most advantageous ways to realize

bearing feature extraction and selection through signal processing technology [6]. When the bearing is degraded, it usually manifests in different degrees in the time field, frequency field, and the combination of the two fields. Therefore, many scholars try their best to extract features of rotating machinery through signal processing technology. Cui et al. [7] proposed an approach to determine whether a mechanical failure had occurred. If a failure had occurred, wavelet denoising on the vibration signal was performed, and features were extracted from the time field and frequency field to break the limit of utilizing merely a single field. Daviu et al. [8] proposed a diagnosis method for the state of the damping rod of the synchronous motor by using the EMD method. By analyzing the motor stator current, it could track the characteristic transient evolution of specific faults of related components in the time-frequency diagram. Mba et al. [9] proposed a new fault detection system based

on the combination of stochastic resonance and the hidden Markov model (HMM). It used stochastic resonance noise to amplify the weak pulse and used HMM method to model the system observation data into the probability function of the hidden state of the system. Wang et al. [10] adopted the KPCA approach to acquire the covariables of the Weibull proportion hazard model and predicted the RUL of roller bearings with great precision. Miao et al. [11] proposed an improved maximum correlated kurtosis deconvolution (MCKD). This method estimated the iteration period by calculating the autocorrelation of the envelope signal, rather than relying on the given previous period. This method had a good effect on bearing analysis under harsh working conditions. Li et al. [12] proposed a new hybrid modeling method, which used the variational modular decomposition method to combine time series decomposition, feature selection, and basic prediction model into a synchronous optimization framework. The hybrid modeling method had achieved good results in wind speed prediction. These methods have achieved good results based on signal processing technology, but it required experts to deeply understand the operation mechanism of rotating machinery, master signal processing technology, and design feature extraction methods to realize the degradation life prediction of rotating machinery.

In recent years, deep learning is obviously superior to the traditional shallow learning methods in adaptive feature learning ability and multilayer nonlinear mapping ability [13]. Therefore, the research of applying it to the prediction of mechanical residual life presents a "blowout" development. Zhao et al. [14] introduced the application of various in-depth learning methods in machine health monitoring and verified them through experiments. All methods achieved good monitoring results. Li et al. [15] extracted the time-frequency domain information of bearing through a multiscale convolution neural network to realize life prediction, which had high prediction accuracy. Jiang et al. [16] proposed a method for predicting the remaining useful life of bearings based on the combination of time series multichannel convolutional neural network (CNN) and attention-based long-term and short-term memory network (LSTM). This method divided the time series into multiple channels and improved the performance by using CNN, LSTM, and the attention-based mechanism method. Qin et al. [17] proposed a new neural network method based on a gated dual attention unit for the prediction of RUL of rolling bearings, which used rolling bearing life-cycle vibration data to calculate a series of root mean squares at different times as the health indicator vector, and estimated the remaining life accurately by predicting the health indicator vector. Verstraete et al. [18] proposed a depth countermeasure semisupervised method based on multisensor fusion to predict RUL. This method has good prediction ability in turbofan engine and rolling bearing. Ellefsen et al. [19] used a semisupervised deep structure to predict the remaining service life of turbofan engine degradation. In addition, a genetic algorithm (GA) method was applied to adjust a large number of hyperparameters in the training process. Zhu et al. [20]

combined wavelet transform and CNN to predict bearing RUL. Firstly, wavelet transform was used to extract time-frequency features, and then, multiscale CNN was used to estimate bearing RUL. Compared with signal processing technology, these methods avoid the complex feature extraction work, directly input the data in the time domain or frequency domain into various neural networks for bearing RUL, and have good results. However, these methods need a large number of samples to train the model; otherwise, it will lead to the prediction error or overfitting of the depth model [21].

In the actual manufacturing process, the bearing is in normal operation in most cases, so many bearing degradation sample data are not easy to obtain. For the study of bearing degradation under limited samples, on the one hand, the model can be improved. Such as Xiang et al. [22] combined the time domain and frequency domain features of gear vibration signals to generate gear health indicators and proposed a new type of LSTM neural network with weight amplification (LSTMP-A) predicting gear remaining useful life. This method can be weighted according to the contribution degree of input data, so as to make better use of a few samples to predict RUL. Xiang et al. [23] also proposed a multicellular LSTM-based deep learning model, which used hierarchical division units to determine the importance of input data, centrally retain the global trend, timely update the local trend, and more effectively mined the degradation trend of different degrees in limited samples. The above methods train models from a few samples through the design of complex architecture and loss function, which has the problem of overfitting the data that it can reduce the generalization ability and application scope of the model. On the other hand, the data can be enhanced by using methods of data generation. Such as the field of image recognition, many scholars improve the recognition capability of the model by augmenting the information set through the generative adversarial networks (GAN) [24]. For example, Zhong et al. [25] transferred the labeled training image style to each camera by using CycleGAN and formed an enhanced training set together with the original training samples. The model trained by the enhanced training set had a good effect. Oliveira et al. [26] used GAN to generate high-quality data to increase the classification accuracy of viseme. Huang et al. [27] used GAN to achieve crossdomain adaptive data augmentation methods. Inspired by the above methods, when bearing data are not easy to obtain, we can use GAN to produce bearing degeneration information to augment the data set.

However, GAN still has many drawbacks, such as training instability, training failure, gradient disappearance, and mode collapse. To overcome these shortcomings, researchers have designed some GAN variants. CGAN [28] is an improved architecture with conditional labels, which enables GAN to achieve better model convergence and the ability to avoid model collapse than traditional GANs. WGAN [29] and WGAN-GP [30], which added gradient penalty term, used Wasserstein distance to replace Jensen-Shannon divergence of original GAN, greatly improved the stability of model training. Furthermore, to tackle the

problems of bearing remaining useful life prediction and the difficulty of obtaining training data samples, a new framework is proposed in this paper, including a data enhancement method combining CGAN and WGAN (CWGAN) and a bearing RUL prediction model based on deep adversarial convolutional neural network (AdCNN). In the data augmentation method (CWGAN), the obtained partial degraded bearing samples are input into the CWGAN network for antagonistic training. When the network reaches Nash equilibrium, a large number of bearing degradation samples are generated from the generator and then mixed into the existing samples to train the AdCNN prediction model. In the prediction model (AdCNN), the obtained original vibration signal is first subjected to fast Fourier transformation to get the frequency domain signal, and then, it is input into the AdCNN model for adversarial training. After adaptively extracting bearing degradation features layer by layer through 1DCNN, the bearing RUL is predicted. Then, the prediction results use exponential smoothing [31] to reduce the problem of their local volatility. The experiment results show that the AdCNN model for bearing RUL prediction exhibits high prediction accuracy, and the generated sample data based on CWGAN has significant data augmentation capabilities. Therefore, the bearing RUL prediction method combined with CWGAN and AdCNN can achieve effective RUL prediction under few samples through high-quality CWGAN-generated data. The main work of our paper is

- (i) Combine the one-dimensional CNN and the adversarial neural network, use adversarial training to realize the prediction of bearing RUL, carefully adjust the framework of the convolution layer and pool layer in the network, and improve the RUL prediction precision of AdCNN for the bearing. Batch normalization and dropout techniques are applied in the predictor to accelerate adversarial model training, solve the vanishing gradient problem, and avoid overfitting.
- (ii) Under the condition of few samples, the CWGAN combining WGAN and CGAN has better stability, model convergence, and prediction ability to avoid mode collapse. It can generate a lot of high-quality data to augment the dataset to improve the precision and adaptability of the bearing RUL prediction pattern.
- (iii) The index smooth approach is adopted to process the prediction results of the bearing, which remarkably reduces the volatility of the prediction results and further improves the global prediction accuracy.

The other parts of our research are arranged as demonstrated. Section 2 introduces the basic theory of the approach adopted in this paper. Section 3 mainly introduces the details of the approach that we propose. Section 4 verifies the approach by experiments and gives performance evaluation and comparison outcomes. Eventually, Section 5 is the conclusion.

2. Basic Theory

2.1. Generative Adversarial Networks and Its Variants. GAN mainly consisted of two parts, including the generator and the discriminator [24]. The generator mainly makes the data generated by itself more real by learning the distribution of real sample data, while the discriminator is used to distinguish the authenticity of the received data. In the training process, the generator tries to generate more real data to deceive the discriminator, while the discriminator tries to distinguish the true and false data. After many rounds of the game between generator and discriminator, they will reach Nash equilibrium. So far, the data generated by the generator are close to the distribution of real sample data, and the discriminator cannot correctly identify the authenticity of the data. The structure diagram of GAN is shown in Figure 1.

In the GAN training process, the generator input is a set of random noise $Z = (z^1, z^2, \dots, z^m)$, and the output is the generated sample $G = (G(z)^1, G(z)^2, \dots, G(z)^m)$, which is similar to the real sample distribution. The input of the discriminator is the generated sample $G(Z)$ or the real sample $X = (x^1, x^2, \dots, x^m)$, and the output is a probability value adopted to discriminate the real sample from the produced sample. The discriminator and generator are alternately trained to reach the Nash balance, and the lost function is

$$\min_G \max_D L(D, G = E_{x \sim P_{\text{data}}(x)}[\log(D(x))] + tE_{z \sim P_z(z)}n (\log(1 - D(G(z))))). \quad (1)$$

A conditional generative countermeasure network (CGAN) is an extension of the original GAN. The generator and discriminator add additional information y as condition. Y can be any information, such as category information or other modality information [28]. In the generative pattern, previous input noise $P_z(z)$ and condition information y jointly create a joint hidden tier representation. The adversarial training framework is quite flexible in the composition of a hidden layer representation. Similarly, the objective function of conditional GAN is a two-player minimax game with conditional probability.

$$\min_G \max_D L(D, G) = E_{x \sim P_{\text{data}}(x)}[\log(D(x|y))] + E_{z \sim P_z(z)}[\log(1 - D(G(z|y)))]). \quad (2)$$

Since the instability of the original GAN training process, it is easy to cause problems such as gradient disappearance and model collapse. WGAN is applied and skillfully solves these problems [29]. WGAN introduces Wasserstein distance to solve the problem that JS divergence (Jensen-Shannon divergence) in original GAN is difficult to train due to less coincidence between noise data and real data. WGAN has realized the following progress: (1) the sigmoid activating function of the last tier of the discriminator is discarded; (2) the loss of the generator and discriminator is no longer logarithmic; (3) momentum and momentum are no longer used when optimizing the

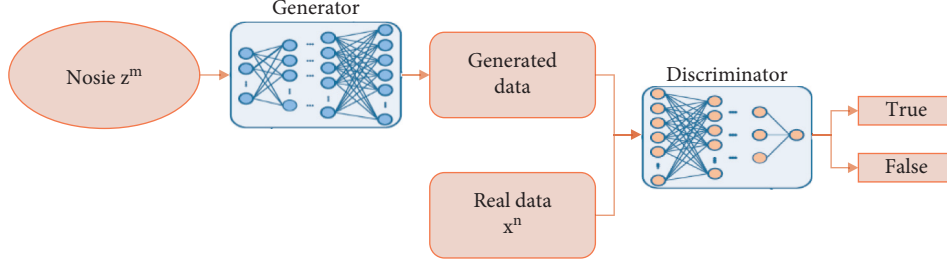


FIGURE 1: The structure diagram of GAN.

network. Adam and other optimization algorithms use RMSProp and SGD; (4) WGAN limits the absolute value of the discriminator to a fixed constant c after each parameter update. The loss function of WGAN with penalty term is

$$V(G, D) = \max_D \left\{ E_{x \sim P_{\text{data}}(x)} [D(x)] - E_{z \sim P_z(z)} [D(x)] - \lambda E_{x \sim P_{\text{penalty}}(x)} \left[(\nabla_x D(x) - 1)^2 \right] \right\}, \quad (3)$$

where $\lambda E_{x \sim P_{\text{penalty}}(x)}(\cdot)$ is the penalty term, which aims to make $D(x)$ smooth enough to make the model converge [30], and $P_{\text{penalty}}(x)$ is the data between the actual data $P_{\text{data}}(x)$ and the produced data $P_z(z)$. $\nabla_x D(x)$ is the derivative of x .

During WGAN training, we no longer have to work hard for the balance between the generator and the discriminator. As long as the discriminator is trained better, the samples generated by the generator are more realistic. Besides, even if the network structure is relatively simple, WGAN can still show good results and avoid the problem of model collapse [29]. In WGAN, we can also identify the quality of the produced sample by the loss function of the discriminator, that is, the Wasserstein distance between the generated sample and the real sample.

2.2. Convolutional Neural Network. CNN is a model composed of neural networks with multiple layers, mainly composed of multiple filtering layers and a classification layer [32]. The filter mainly extracts features from the input data through the convolution layer and the pooling layer. A classifier serves as a multitier sensor comprising some completely connected tiers. CNN serves as a neural net model with multiple levels, which mainly comprises many filtering stages and a classification stage. The filter abstracts characteristics from the input, and it has two levels, namely the convolution layer and the pool layer. The classification stage is a multitier sensor consisting of some fully connected layers (FCLs). The structural model of CNN is shown in Figure 2.

Conv_1 is a convolution layer, pool_1 is a pool layer, and the two take turns to extract two-dimensional data features. Posterior to some convolution layers and pooling layers, all the characteristics after convolution are expressed through an FCL, and then, a classifier is used to accurately classify the sample data.

2.2.1. Convolutional Layer. The convolution layer convolves the input local region with the convolution kernel, and then, the activating unit generates output characteristics. Each filter adopts the identical kernel to extract the local characteristics of the input local region, which is weight sharing. A filter corresponds to a feature map in the following tier, and the quantity of feature maps is known as the depth of the tier. Assuming that the l layer in the CNN network is a convolutional layer, and i denotes the i -th filtering kernel of the l layer, the calculation formula for the l -th layer is shown as follows:

$$y_i^{l+1}(j) = X^l(j) * K_i^l + b_i^l. \quad (4)$$

In the formula, K_i^l represents the shared weight of the i -th convolution kernel in the l -th convolutional layer, b_i^l represents the corresponding offset value of the convolution kernel, and $X^l(j)$ represents the j -th local field of view area in the l -th layer. $y_i^{l+1}(j)$ denotes the j -th characteristic map of the $(l+1)$ -th tier. After the convolution operation is over, the obtained result is inputted into the nonlinear activating function to get the output of this layer of neurons. The activating function used is the RELU function, and its expression is shown in the following formula:

$$\text{RELU}(z) = \begin{cases} z, & z > 0, \\ 0, & z \leq 0. \end{cases} \quad (5)$$

2.2.2. Pooling Layer. In order to decrease the space dimension of the characteristic map and prevent overfitting, a pool layer is usually added to the convolution layer. Maximum pooling is the most commonly used type of pooling. It only takes the most important part of the input, that is, the maximum value. Its expression is shown in the following formula:

$$F_i^{l+1}(j) = \max_{\substack{(j-1)w+1 < t_x \leq jw \\ (j-1)w+1 < t_y \leq jh}} \{H_i^l(t)\}, \quad (6)$$

where $H_i^l(t)$ denotes the value of neuron t in the i -th feature map of the l -th tier, and w and h represent the width and height of the pooling window, respectively. $F_i^{l+1}(j)$ represents the output result posterior to pooling.

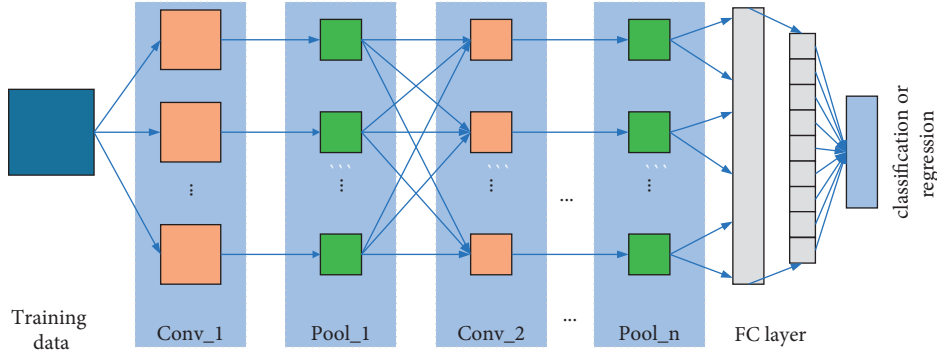


FIGURE 2: Schematic diagram of CNN structure.

2.2.3. Fully Connected Layer. The FCL is like a multilayer perceptron. Its main function is to convert the 2D characteristic map output into a 1D vector, which is convenient for classification and recognition and regression prediction. In classification and recognition, The Softmax classifier is usually used for classification, and its expression is shown in formula (7), where k denotes the quantity of classification species and $\theta^j x$ denotes the parameters of the classification tier. In regression prediction, the mean squared error is often used as the lost function, and its expression is shown in formula (8). In the formula, Y_i denotes the predicted sample value and y_i denotes the real value. M denotes the sum of samples.

$$O_j = \begin{bmatrix} p(y=1)|x; \theta \\ p(y=2)|x; \theta \\ \dots \\ p(y=k)|x; \theta \end{bmatrix} \quad (7)$$

$$= \frac{1}{\sum_{j=1}^k \exp(\theta^j x)} \begin{bmatrix} \exp(\theta^1 x) \\ \exp(\theta^2 x) \\ \dots \\ \exp(\theta^k x) \end{bmatrix},$$

$$MSE = \frac{\sum_{i=1}^n (Y_i - y_i)^2}{M}. \quad (8)$$

2.3. Exponential Smoothing. Exponential smoothing is a method of predicting by calculating the weighted average of past observations [31]. It mainly includes one exponential smoothing, two exponential smoothings, and three exponential smoothings. This paper uses exponential smoothing, which can smooth the time series, eliminate random fluctuations, and find the changing trend of the series. Its calculation formula is shown in the following formula:

$$F_{t+1} = \alpha Y_t + (1 - \alpha)F_t, \quad (9)$$

where Y_t denotes the real observation result in period t , F_t is the predicted value in period t , and α is the smoothing coefficient.

3. Proposed Methodology

Convolutional neural networks have strong characterization capabilities for sequence and image data. Therefore, the AdCNN model combines CNN and GAN to predict the bearing RUL. It changes the generator in GAN to a predictor composed of a 1DCNN. The discriminator mainly comprises a multilayer neural net. The two achieve Nash equilibrium through repeated adversarial training and, finally, use the predictor to calculate the bearing RUL. The main architecture of AdCNN is shown in Figure 3.

3.1. AdCNN Prediction Model Based on Adversarial Training

3.1.1. Flow Diagram. Compared with the original characteristics of the vibration signal, the frequency domain data have a strong regularity and contain more useful information about the original signal, which can help us to quantitatively analyze the vibration signal [33]. The process of the AdCNN prediction approach based on adversarial training is depicted in Figure 4.

Firstly, the vibration signal is obtained from the sensor, each sample $S_i = \{S_i^1, S_i^2, \dots, S_i^n\}$ is transformed by a fast Fourier transform (FFT) and labeled, and the frequency domain samples $X_i = \{X_i^1, X_i^2, \dots, X_i^n, y_i\}$ are obtained, in which i denotes the i -th sample, n represents the dimension of each sample, l_i represents the label corresponding to the sample (l_i is the bearing RUL, and the calculation method is the ratio of the diversity between the present time and the fault time to the difference between the start time of bearing degradation and the time of failure. For instance, if the starting time of a bearing degradation is 1000 seconds, the current time is 1200 seconds, and the bearing failure time is 2000 seconds, and the current remaining useful life of the bearing is $(2000 - 1200)/(2000 - 1000) = 0.8$). Figure 5 shows the original vibration signal waveform and the frequency domain data signal waveform after FFT.

Next, the obtained frequency domain sample data are divided proportionally into training set $X = \{X_1, X_2, \dots, X_m\}$ and test set $T = \{T_1, T_2, \dots, T_m\}$, the

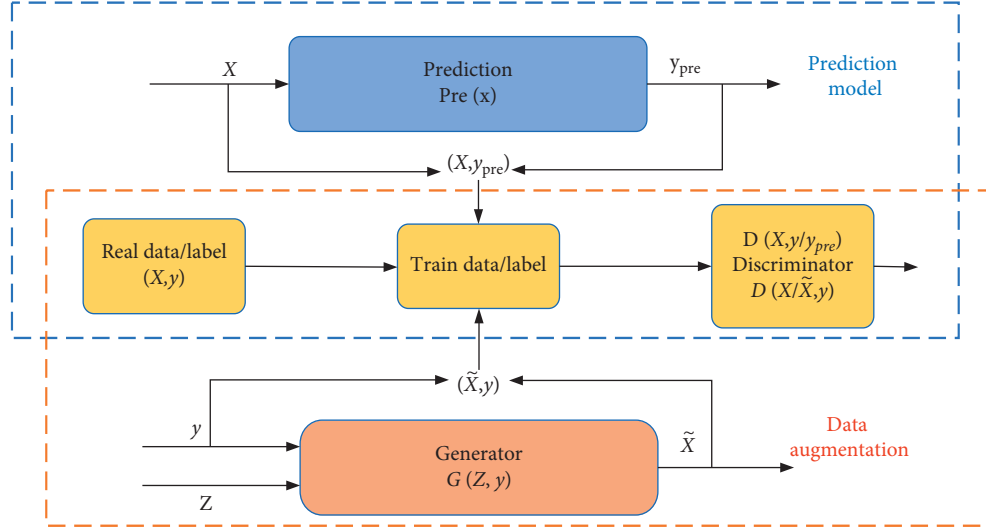


FIGURE 3: General structure of the approach.

AdCNN neural network is built, the network parameters θ are initialized, the learning set is utilized to train the network, after multiple rounds of iterative training, until the converged AdCNN model is obtained, and then, the test set is input to predict the bearing life, and the predicted value \bar{y}_{pre} is obtained.

Finally, the predicted value is smoothed to get the final prediction result y_{pre} .

3.1.2. AdCNN Network Architecture. The main architecture of AdCNN is shown in Figure 6.

The predictor contains a one-dimensional CNN, which comprises filtering and prediction. The filter is used to extract characteristics from the input signal, which is divided into a convolution layer and a pool layer. The prediction is a multilayer perceptron composed of several FCLs. The convolution layer convolutes the input local region with the filter core, and then, the output feature is generated by the activation unit. Each filter uses the same kernel to extract the local features of the input local region, which is weight sharing. In order to reduce the variation of internal covariance and speed up the training process of deep neural network (DNN), batch normalization (BN) layer is introduced. BN layer is usually added after the convolution layer or FCL and before the activation unit. After the convolution operation, the activation function is needed, which enables the network to obtain the nonlinear expression of the input signal, improve the representation ability, and make the learned features more sufficient. We use the popular Rectified linear unit (ReLU) as the activation unit to accelerate the convergence speed of AdCNN. In the AdCNN structure, pool layer is usually added after the convolution layer. As a down-sampling operation, it reduces the space size of network features and the number of neural net parameters. The most commonly used pool layer is the maximum pool layer, which performs a local maximum operation on the input features to reduce the number of parameters and obtain location invariant features.

One part of the discriminator is used to calculate the mean square error between the predicted value (RUL) and the real value (real RUL) of the predictor, and the other part is a multilayer neural network. Its purpose is to learn the matching degree between the input data and its corresponding label. When the input data and label (real RUL) are input into the multilayer neural network, it will give a high score. The detailed network parameter settings of the discriminator will be introduced in the experimental part.

After multiple iterations of the discriminator and the predictor, the discriminating ability of the discriminator is getting more and more potent, and the predicting ability of the predictor is getting stronger and stronger. Finally, the two reach the Nash balance and then take out the predictor alone to predict the remaining life of bearing. The network loss function is shown in the following formula:

$$\begin{aligned} \min_{P, M} \max_D L(D, P) = & E_{x, y \sim P_{data}(x, y)} [\log(D(x, y))] \\ & + E_{x \sim P_{data}(x)} [\log(1 - D(x, P(x)))] \quad (10) \\ & + E_{x, y \sim P_{data}(x, y)} [M(P(x), y)], \end{aligned}$$

where $D(\cdot)$ represents the discriminator, $P(\cdot)$ represents the predictor, $M(\cdot)$ represents the calculated mean square error, x denotes the training sample, and y denotes the relevant training sample label.

3.1.3. AdCNN Training. Now, we discuss how AdCNN is trained. As shown in Algorithm 1, the training process is mainly to optimize the network by updating the weight parameters of the discriminator and predictor through continuous iterations. The update optimization of the discriminator (line3–line8) is mainly realized by increasing the score of matching the sample with the real label, reducing the score of matching the sample with the predicted value, and reducing the mean square error between the real label and the predicted label. The update optimization of the

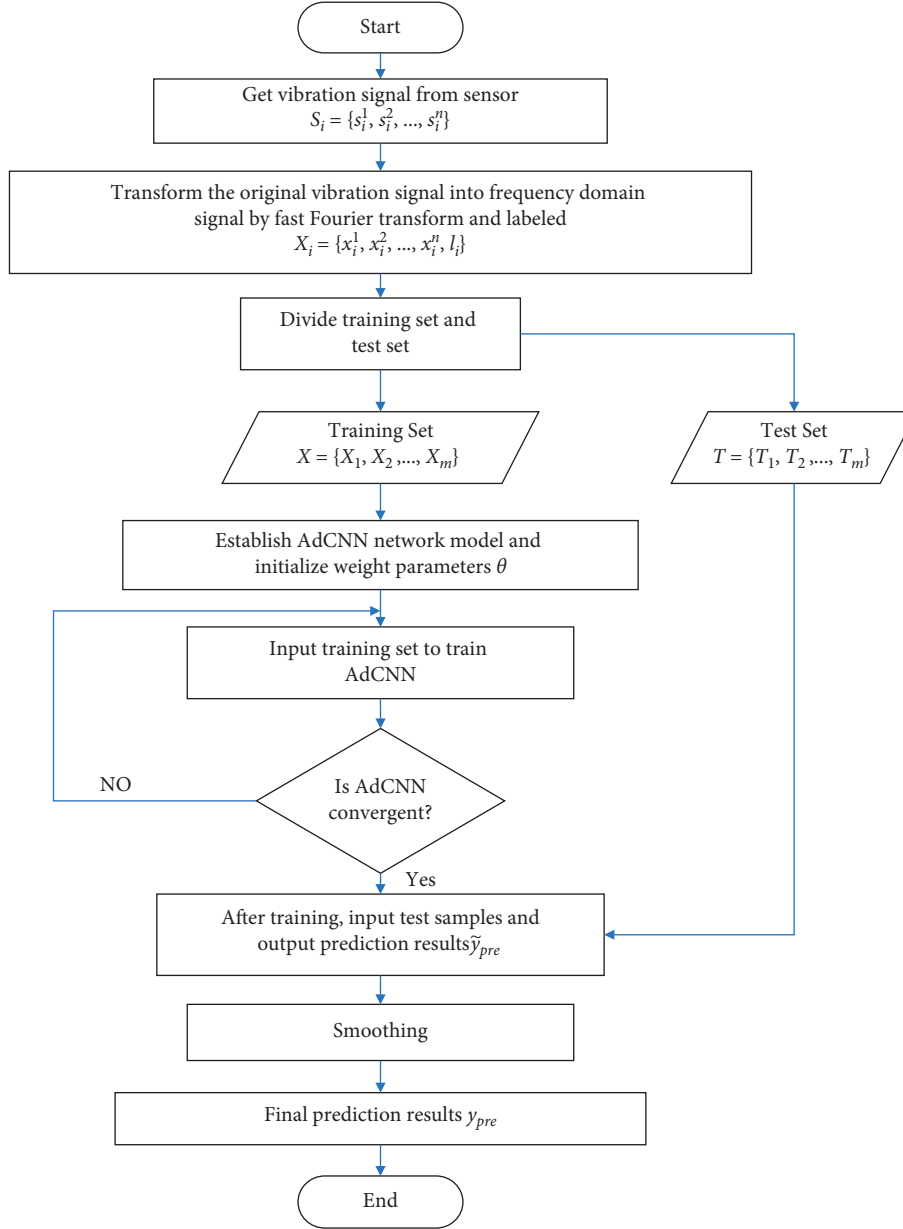


FIGURE 4: Flow diagram of bearing prediction method based on AdCNN.

predictor (line9–line12) is mainly to obtain a higher score in the discriminator by learning to generate a predicted value closer to the real label.

3.1.4. Smoothing. Although deep neural networks can usually obtain a clear global pattern in the estimated RUL, regional undulation unavoidably emerges, which usually leads to undependable prognosis property [15]. To solve the prognosis problem caused by local fluctuations, this paper uses an exponential smoothing method. The global prediction result $\tilde{y}_{pre} = \{\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_m\}$ output by the predictor is exponentially smoothed to obtain the final prediction result $y_{pre} = \{y_1, y_2, \dots, y_m\}$. The specific smoothing method is shown in the following formula:

$$\begin{aligned}
 y_1 &= \tilde{y}_1, \\
 y_2 &= \alpha \tilde{y}_1 + (1 - \alpha) y_1, \\
 y_3 &= \alpha \tilde{y}_2 + (1 - \alpha) y_2, \\
 &\vdots, \\
 y_m &= \alpha \tilde{y}_{m-1} + (1 - \alpha) y_{m-1},
 \end{aligned} \tag{11}$$

where α ($0 < \alpha < 1$) represents the smoothing coefficient. When the prediction sequence has large fluctuations, it is better to choose a larger α , which can quickly keep up with changes; otherwise, it is better to choose a smaller α .

3.2. CWGAN-Based Data Generation Model. In an actual production environment, it is difficult to acquire massive

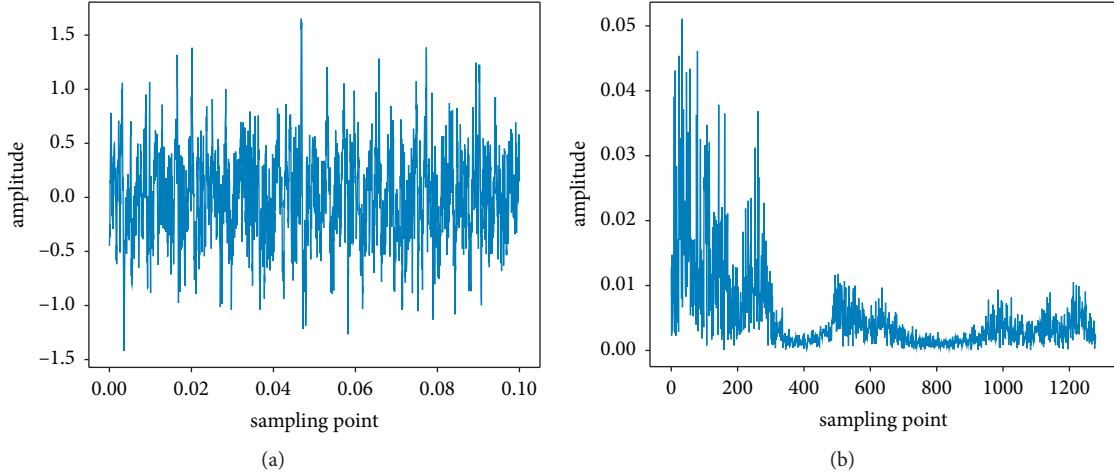


FIGURE 5: Time-field and frequency field signal waveform. (a) Original time-field vibration signal waveform. (b) Frequency domain vibration signal waveform by FFT.

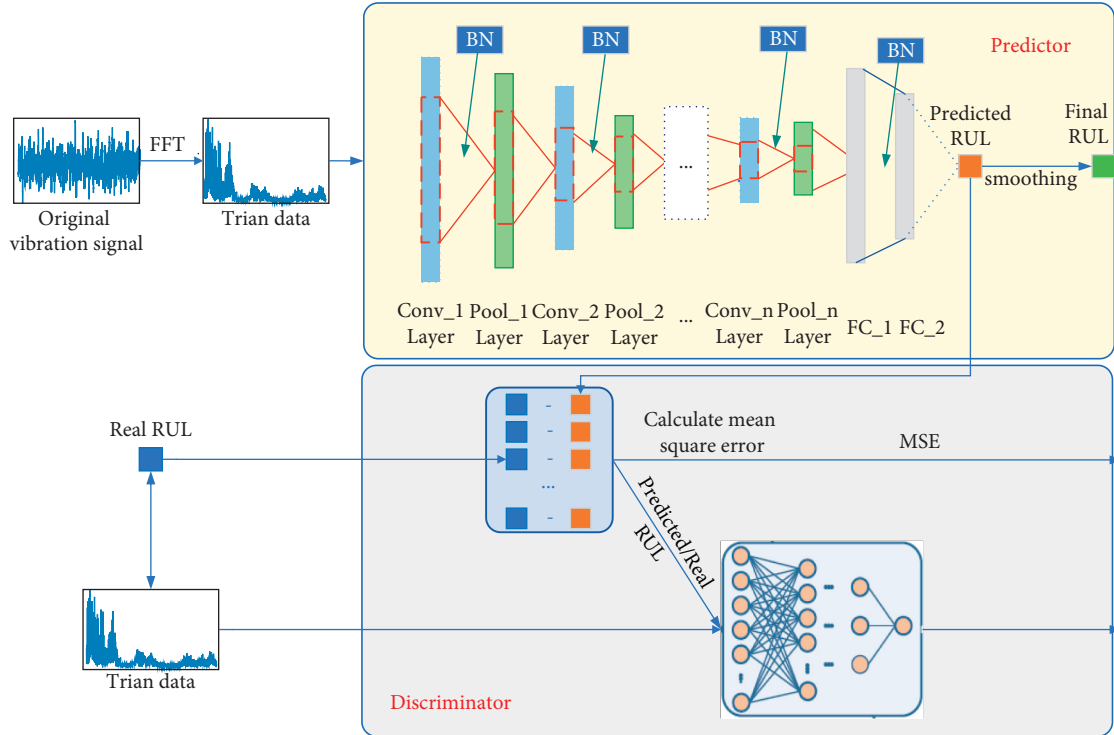


FIGURE 6: Network architecture diagram of AdCNN model.

high-quality sample information. Therefore, this paper proposes a data enhancement method based on CWGAN. The basic idea is to combine the advantages of WGAN and CGAN to generate sample data. The specific flow diagram is shown in Figure 7. The input of the generator is real RUL and noise data, and the output is false sample data; there are two types of discriminator input, one is real RUL and false sample data, and the other is real RUL and real sample data. After multiple iterations of the generator and discriminator, after the network reaches the Nash equilibrium, massive sample information is produced by the generator to enhance

the data set. The loss function of CWGAN is shown in the following formula:

$$V(G, D) = \max_D \{ E_{x, y \sim P_{\text{data}}(x, y)} [D(x, y)] - E_{x, y \sim P_G(x, y)} [D(x, y)] - \lambda E_{x, y \sim P_{\text{penalty}}(x, y)} [(\nabla_x D(x, y) - 1)^2] \}, \quad (12)$$

where $D(x, y)$ is the discriminator and $\lambda E_{x, y \sim P_{\text{penalty}}(x, y)}(\cdot)$ is the penalty term, which aims to make $D(x)$ smooth enough and the model converges, $P_{\text{penalty}}(x, y)$ refers to the data

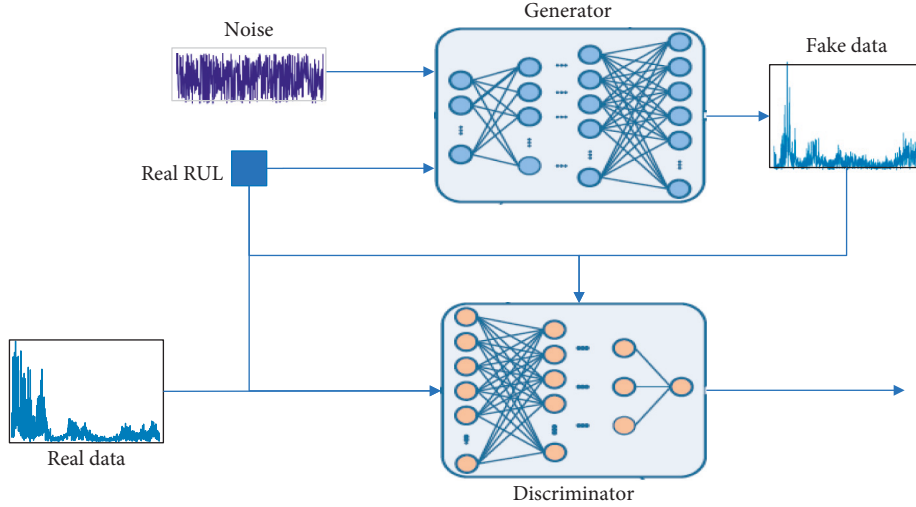


FIGURE 7: Network architecture diagram of CWGAN model.

```

(1) Initialize: discriminator with parameter  $\theta_D$ , predictor with parameter  $\theta_P$ .
(2) for  $k$  training iterations do
(3)   for  $k_d$  iterations do
(4)     specimen  $m$  example  $\{(X_1, y_1), (X_2, y_2), \dots, (X_m, y_m)\}$  from dataset
(5)     obtaining predicted data  $\{X_1, X_2, \dots, X_m\}$ ,  $\tilde{y}_i = P(X_i)$ .
(6)     update  $\theta_D$  by descending along its gradient
(7)      $\nabla_{\theta_D} [1/m \sum_{i=1}^m \log D(X_i, y_i) + 1/m \sum_{i=1}^m \log(1 - D(X_i, \tilde{y}_i)) + 1/m \sum_{i=1}^m (\tilde{y}_i - y_i)^2]$ 
(8)   end for
(9)   for  $k_p$  iterations do
(10)    update  $\theta_P$  by descending along its gradient
(11)     $\nabla_{\theta_P} [-1/m \sum_{i=1}^m \log(D(X_i, P(X_i)))]$ 
(12)  end for
(13) end for

```

ALGORITHM 1: AdCNN training process.

between the real data $P_{\text{data}}(x, y)$, and the generated data $P_G(x, y)$. $\nabla_x D(x, y)$ is the derivative of x .

Now, we discuss how to train CWGAN. As shown in Algorithm 2, the training process still optimizes the network through constant iterations to renew the weight variables of the discriminator and generator. The update optimization of the discriminator (line3–line9) mainly includes increasing the sample to match the real label score and reducing the score that the sample matches the predicted value. The update optimization of the generator (line10–line13) is mainly to obtain a higher score in the discriminator by learning to generate a predicted value closer to the real label.

4. Experiments

The experimental data adopt the bearing degradation data set of the IEEE 2012 PHM data challenge [34]. In this experiment, the threshold of complete degradation of the bearing is set as the amplitude is greater than 20 g. The collected data are from the acceleration sensors in the horizontal and vertical directions, and the sampling frequency is 25.6 kHz. The information collection is carried out every 10 s, each time

duration is 0.1 s. Therefore, there are 2560 vibration acceleration data for each sample data. As shown in Table 1, the data set contains 17 bearing degradation data under three different working conditions based on many literature studies [35, 36], and the horizontal vibration signal can provide more useful degradation information than the vertical signal. Therefore, the experiments in this paper use the horizontal vibration signal as the training data.

4.1. RUL Prediction Experiment Process Based on AdCNN

4.1.1. AdCNN Model Parameter Description. This experiment builds a prediction model based on Google's open-source deep learning framework-Tensorflow. The AdCNN model is mainly composed of predictors and discriminators. The quantity of tiers of the predictor net model is set as depicted in Table 2. The model has 3 convolution layers and 3 pool layers. The first convolutional layer conv_1 has 32 convolution kernels (the kernel size is 21×1), the second conv_2 layer has 48 convolution kernels (11×1), and the third conv_3 has 64 convolution kernels (7×1). The sizes of

```

(1) Initialize: discriminator with parameter  $\theta_D$ , generator with parameter  $\theta_G$ .
(2) for  $k$  training iterations do
(3)   for  $k_d$  iterations do
(4)     sample  $m$  example  $\{(X_1, y_1), (X_2, y_2), \dots, (X_m, y_m)\}$ . from dataset
(5)     sample  $m$  noise samples  $\{Z_1, Z_2, \dots, Z_m\}$  from the prior  $P_{\text{prior}}(Z)$ 
(6)     obtaining generator data  $\{\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_m\}$ ,  $\tilde{X}_i = G(Z_i, y_i)$ 
(7)     update  $\theta_D$  by descending along its gradient
(8)      $\nabla_{\theta_D} [1/m \sum_{i=1}^m D(X_i, y_i) - 1/m \sum_{i=1}^m (D(\tilde{X}_i, y_i))] + \lambda E_{x,y \sim P_{\text{penalty}}(x,y)} [(\|\nabla_x D(x, y)\| - 1)^2]$ 
(9)   end for
(10)  for  $k_p$  iterations do
(11)    update  $\theta_G$  by descending along its gradient
(12)     $\nabla_{\theta_G} [-1/m \sum_{i=1}^m D(G(Z_i, y_i), y_i)]$ 
(13)  end for
(14) end for

```

ALGORITHM 2: CWGAN training process.

TABLE 1: The details of bearing dataset.

Conditions	Load (N)	Velocity (rpm)	Bearings			
1	4000	1800	Bearing1_1	Bearing1_2	Bearing1_3	Bearing1_4
			Bearing1_5	Bearing1_6	Bearing1_7	
2	4200	1650	Bearing2_1	Bearing2_2	Bearing2_3	Bearing2_4
			Bearing2_5	Bearing2_6	Bearing2_7	
3	5000	1500	Bearing3_1	Bearing3_2	Bearing3_3	

TABLE 2: Details of the predictor networks in this experiment.

Layer type	Kernel	Channel	Stride	Activation	Number of training parameters
Input	—	1	—	—	—
Conv_1	21×1	32	2	ReLU	704
Pool_1	2×1	32	1	—	—
Conv_2	11×1	48	2	ReLU	576
Pool_2	2×1	48	1	—	—
Conv_3	7×1	64	2	ReLU	512
Pool_3	2×1	64	1	—	—
FC_1	—	1	—	ReLU	4981248
FC_2	—	1	—	ReLU	65664
Output	—	1	—	Sigmoid	257

the three pooling layers are 4×1 , 2×1 , and 2×1 . There are two FCLs FC_1 and FC_2, which are composed of 512 and 256 neurons, respectively. In order to avoid overfitting of the pattern, dropout technology is adopted in the FCL with a dropout ratio of 0.7. The pooling layers pool1, pool2, and pool3 adopt the maximum pooling method. The first few layers in the network use the ReLU activating function, and the eventual output tier is the sigmoid activating function, because the bearing RUL is at a value of 0-1. The learning rate of the predictor is $lr_P = 0.001$, the batch scale is 100, and the iterations are 3000.

The discriminator network model has five FCLs, and the three hidden layers are composed of 64, 128, and 256 neurons, respectively. The parameter settings are shown in Table 3. The batch size during model training is 100, the learning rate (lr_D) is 0.0001, and the iterations are 15000 times. During

the training process, we trained the predictor P once after completing the training of discriminator D for 5 times.

4.1.2. Experimental Process and Result Analysis. In this experiment, 16 bearing data from all the bearings shown in Table 1 will be selected as a training set, and the rest one that Bearing2_7 will be used as test set. To evidence the superior performance of our model, DNN [37], LSTM [38], and SVR [39] are designed and compared with the methods in this paper. Among them, DNN is to input the time-frequency features compressed by self-encoding into the deep neural network for bearing life prediction. The parameter setting is that the neural network has 9 layers, the activation function of the hidden layer is ReLU, the activation function of the output layer is sigmoid, the epoch is 50, the learning rate is

TABLE 3: Details of the Discriminator networks in this experiment.

Layer	Dimension	Activation	Number of training parameters
Input	1281	—	—
Hidden_1	64	ReLu	82084
Hidden_2	128	ReLu	8320
Hidden_3	256	ReLu	33024
Output	1	Sigmoid	257

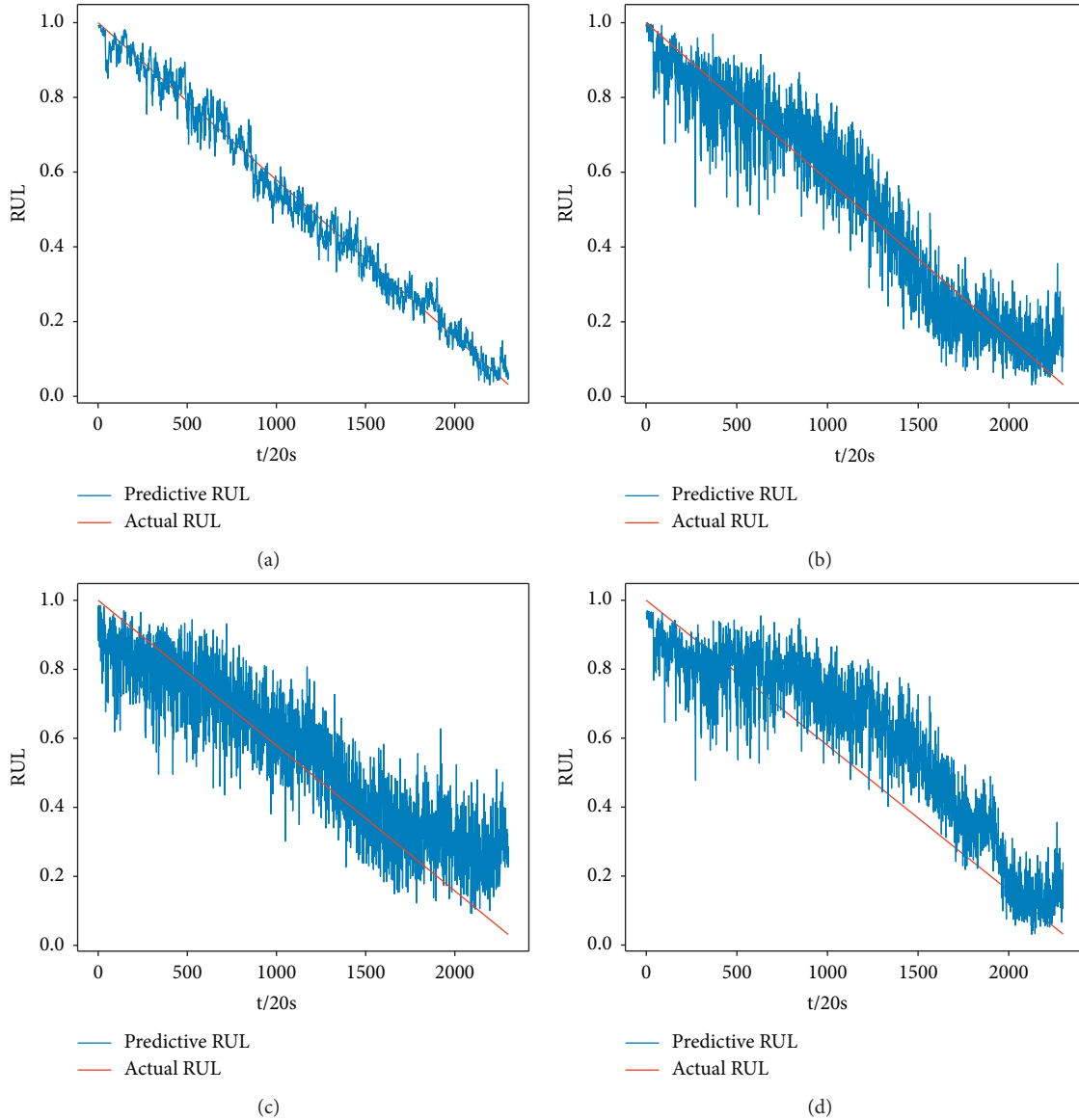


FIGURE 8: The remaining life prediction outcomes of different methods in the same test bearing. (a) AdCNN's remaining life prediction results; (b) LSTM's remaining life prediction results; (c) DNN's remaining life prediction results; (d) SVR's remaining life prediction results.

0.001, the optimization algorithm adopts SGD, and the loss function is the mean square error. The LSTM model first extracts the bearing characteristic parameter set (frequency domain root mean square, time-domain root mean square, frequency domain average amplitude, time field peak-to-peak value, frequency field variance, wavelet packet third

frequency range standardized power spectrum, wavelet packet first 7 band normalized energy spectrum) and inputs the characteristic parameters into the LSTM network for bearing life prediction. The parameters setting is that the LSTM has 4 layers, hidden units are 240, time step is 40, the batch size is 40, and the learning rate is 0.0006. The bearing

performance degradation indicators in SVR are root mean square and kurtosis, and the kernel function uses the RBF kernel function. The kernel function selects the ‘‘RBF’’ kernel, the penalty factor is 325, and the kernel function parameter is 0.06. All these models have experimented with the bearing training set and test set mentioned. The learning set is characterized by corresponding feature extraction methods and then inputs into the corresponding net to obtain the corresponding prediction results. The prediction results of the 4 models on the same test bearing are shown in Figures 8(a)–8(d).

It can be seen from (a)–(d) in Figure 8 that the proposed method has a high degree of fit with the predicted value of real bearing degradation life, indicating that it has good prediction ability. As shown in Figure 8(a), AdCNN makes the distribution of the model closer to the real distribution by using adversarial training, so the error of the predicted RUL result is the smallest, and the local volatility problem is reduced by exponential smoothing. As shown in Figures 8(b)–8(d), it can be observed that the RUL prediction results of the other three methods have greater errors and more obvious volatility compared with the real RUL values. This indicates that AdCNN has a better ability to predict bearing RUL. In addition, compared with SVR, the prediction effects of the other three methods have better performance, which reveals that the deep learning approach performs better for bearing RUL prediction. In order to quantify the prediction errors of the four methods, we use the average absolute error (formula (13)), the root mean square error (formula (14)), and the maximum error (formula (15)) to measure the prediction effects of various methods. The different errors corresponding to the three methods are shown in Table 4.

$$\text{MAE} = \frac{\sum_{i=1}^m |y_{\text{pre}}^i - y^i|}{m}, \quad (13)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^m (y_{\text{pre}}^i - y^i)^2}{m}}, \quad (14)$$

$$\text{ME} = \max\{|y_{\text{pre}}^i - y^i|\}. \quad (15)$$

From the error statistics of Table 4, it can be seen that the average absolute errors of AdCNN compared with SVR, DNN, and LSTM methods are reduced by 77.21%, 66.30%, and 51.56%, and the root means square error has been reduced by 64.81%, 51.69%, and 33.72%, respectively. For the maximum error, AdCNN is the smallest, and the other three methods are relatively large, which further shows that AdCNN has less volatility. The quantized error statistics confirm the experimental results in Figure 8. Obviously, compared with the other three methods, AdCNN has achieved the best performance in the above three quantitative indicators, which proves the effectiveness of AdCNN in bearing RUL life prediction. In addition, the three deep learning methods are less than SVR in various errors, which fully reflects the

TABLE 4: Comparison of various error values of different methods.

Method	Error type		
	MAE	RMSE	ME
SVR	0.136	0.162	0.51
DNN	0.092	0.118	0.45
LSTM	0.064	0.086	0.37
AdCNN	0.031	0.057	0.13

robustness and generalization ability of the deep learning method.

4.1.3. Ablation Study. In this part, we mainly focus on the design of AdCNN. We conduct several ablation studies by removing or replacing one design component from AdCNN at a time. We mainly divided into the following situations:

- (i) 1DCNN: In this model, only one-dimensional CNN is adopted for bearing RUL prediction. The net structure and hyperparameters are consistent with those of the predictor in AdCNN. The model loss function only uses the mean square error.
- (ii) AdDNN: In this model, we employ a simple deep neural net to replace the predictor in AdCNN, and the discriminator module is consistent with the model proposed in this paper.
- (iii) AdRNN: In this model, a cyclic neural net is adopted to replace the predictor in AdCNN, and the rest of the network structure is consistent with AdCNN.
- (iv) AdCNN_nos: In this model, there is no exponential smoothing for the bearing prediction RUL generated by the network, and the network output value is directly used as the predicted RUL. The overall network structure is consistent with the AdCNN mentioned in this paper.

Figure 9 shows the box plot of the absolute error (calculated as $AE = |y_{\text{pre}} - y|$) between the RUL y_{pre} and the real RUL y of the predicted bearing of various architectures. The box graph contains the information of quartile and dispersion degree, which can easily compare the effects of different modules. Compared with AdCNN and 1DCNN, the absolute error of AdCNN is significantly less than that of 1DCNN, which shows that adding a discriminator to 1DCNN will have a better prediction effect. Comparing AdCNN and AdCNN_nos, the absolute error interval of AdCNN_nos is larger, which shows that exponential smoothing has a certain effect on large data fluctuations. In addition, AdCNN has a smaller absolute error distribution compared with AdDNN and AdRNN, which further shows that AdCNN has a higher predictive ability for bearing RUL.

4.2. Data Augmentation Experiment Process Based on CWGAN

4.2.1. CWGAN Model Parameter Description. Cwgan model is mainly composed of generator and discriminator. The

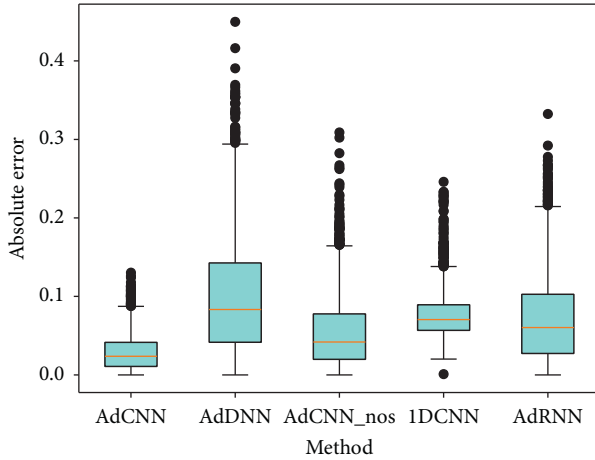


FIGURE 9: Box plot of prediction absolute error in various model situations.

number of layers of its network model is shown in Table 3. The generator has an input dimension of 101 (100-dimensional noise and 1-dimensional label), 128 neurons in the first hidden layer, 200 neurons in the second hidden layer, and an output dimension of 1280, representing the generated samples. The input dimension of the discriminator is 1281 (1280-dimensional sample data and 1-dimensional label). The first hidden layer has 128 neurons, and the second hidden layer has 256 neurons. The output is one-dimensional, indicating the true and false probability of generating sample data. The hidden layer of generator and discriminator adopts the ReLU activation function, the last layer of generator adopts the sigmoid activation function, and the last layer of discriminator has no activation function. Generator learning rate $LR_G = 0.0001$, discriminator learning rate $LR_D = 0.00005$, batch size = 128, and the number of iterations is 30000.

The CWGAN model consists of a generator and a recognizer, and the number of layers of its network model is set as shown in Table 5. The generator has an input dimension of 101 (100-dimensional noise and 1-dimensional label), the first hidden tier has 128 neurons, the second hidden tier has 200 neurons, and the output dimension is 1280, representing the generated samples. The input dimension of the discriminator is 1281 (1280-dimensional sample data and 1-dimensional label), the first hidden layer carries 128 neurons, the second hidden layer has 256 neurons, and the output is one-dimensional, indicating whether the generated specimen data are true or false probability. Both the generator and the discriminator hidden layer adopt the ReLU activation function, the last layer of the generator adopts the sigmoid activating function, and the last layer of the discriminator has no activation function. The study rate of the generator lr_G is 0.0001, the study rate of the discriminator lr_D is 0.00005, the batch scale is 128, and the iterations are 30000.

4.2.2. Experimental Process and Result Analysis. In this experiment, we only take 40% of the real data in Table 1 for adversarial training. This is to simulate the actual situation

TABLE 5: Details of the CWGAN networks in this experiment.

Layer	Dimension	Activation	Number of training parameters
G_input	101	—	—
G_h1	128	ReLu	13056
G_h2	200	ReLu	25800
G_output	1280	Sigmoid	257280
D_input	1281	—	—
D_h1	128	ReLu	163968
D_h2	256	ReLu	33024
D_output	1	—	512

with fewer high-quality data samples. Figures 10(a)–10(c), respectively, show the frequency domain waveforms of real samples and the frequency domain waveforms of samples generated by CWGAN under three operation environments. The data generated by the CWGAN model under conditions 2 and 3 are quite similar to the real data. Although the generated data under condition 1 are slightly different in some frequency bands, it can still be seen that the overall distribution of generated data is basically consistent with the original frequency domain data. Intuitively, it shows that CWGAN has sufficient representation ability that the generated high-quality samples can be used to enhance the data set to alleviate the problem of few samples in the actual situation.

In the case of few original samples, adding these CWGAN-generated samples will make the data sample features richer, and training the mixed samples can greatly improve the prediction accurateness and pattern generalizability.

To verify the effectiveness of enhancing the data set through CWGAN, a large number of bearing degradation samples are generated from the generator and then mixed into the existing samples to train the AdCNN prediction model. When the converged AdCNN model is obtained, the test set is input to predict the bearing life. This experiment is to ensure that the specimen quantity in the training data collection does not change. The original data and the CWGAN-generated data in the training data set are set at different ratios: 90%–10%, 80%–20%, 60%–40%, 40%–60%, 20%–80%, 10%–90%. Input different proportions of training data into the three methods (DNN, SVR, and LSTM) are mentioned in Section 4.1.2 and AdCNN to train the prediction model. The same bearing data were then used to test the prediction accuracy of the four models.

Table 6 shows the mean absolute deviation and root average squared deviation of the RUL predicted by the training model under different percentages of the generated data in the learning set. For the convenience of observation, data in Table 6 are drawn in the line graph, which is shown in Figure 11. As it is presented in Table 6 and Figure 11, AdCNN has the highest prediction accurateness of the model trained under different proportions of generated data. Observing the line graph, we can find that as the proportion of generated data continues to increase, the prediction accuracy of various methods is different. When declining the generated data accounted for 10%–60% of the training data, the MAE and RMSE of DNN, LSTM, and AdCNN did not rise significantly. This shows that when the generated data

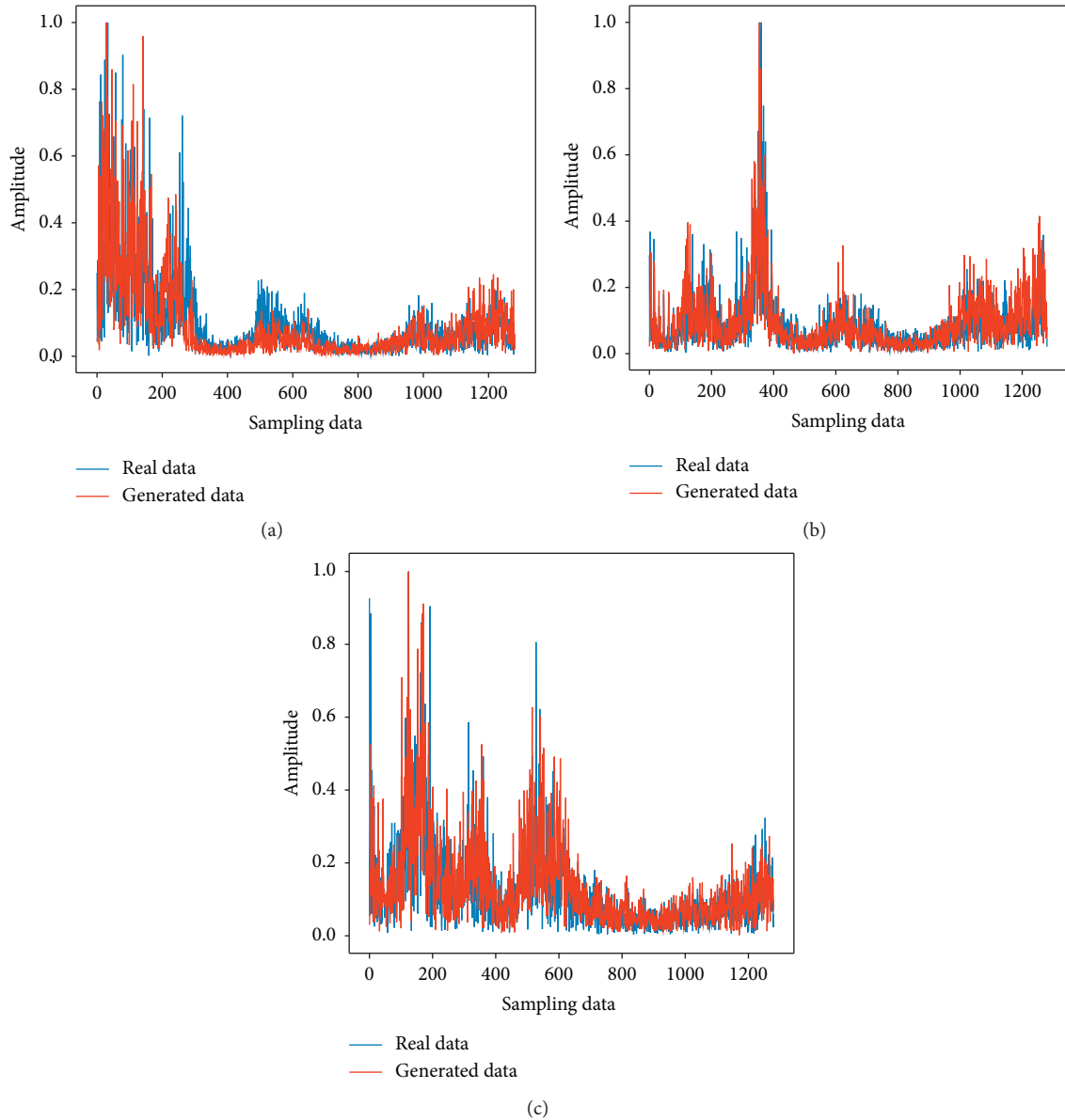


FIGURE 10: Frequency domain waveform diagram of real data and generated data under different working conditions. (a) Frequency domain waveform diagram under working condition 1. (b) Frequency domain waveform diagram under working condition 2. (c) Frequency domain waveform diagram under working condition 3.

TABLE 6: MAE and RMSE of different percentages of generated data training model prediction results.

Methods		Percentage of generated data					
		10%	20%	40%	60%	80%	90%
SVR	MAE	0.142	0.156	0.232	0.275	0.247	0.223
	RMSE	0.163	0.174	0.251	0.243	0.269	0.251
DNN	MAE	0.095	0.102	0.097	0.113	0.158	0.212
	RMSE	0.117	0.121	0.109	0.136	0.173	0.229
LSTM	MAE	0.067	0.065	0.071	0.092	0.105	0.127
	RMSE	0.085	0.082	0.094	0.104	0.122	0.143
AdCNN	MAE	0.038	0.029	0.039	0.074	0.085	0.092
	RMSE	0.051	0.046	0.058	0.092	0.097	0.105

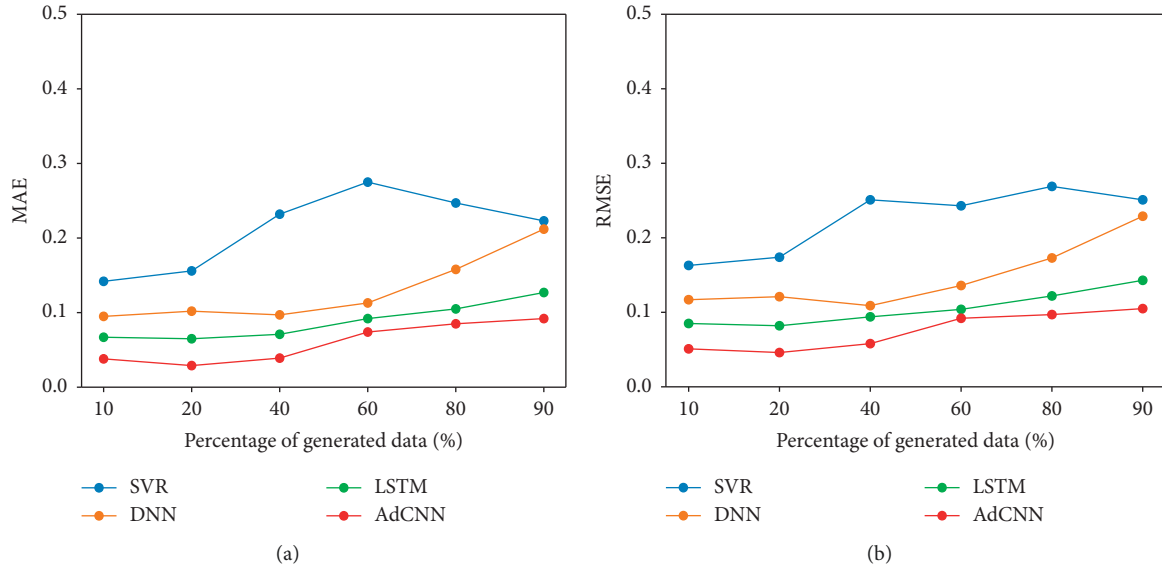


FIGURE 11: Line graphs of MAE and RMSE in different percentages of generated data training model prediction results. (a) MAE of model prediction results. (b) RMSE of model prediction results.

accounted for less than 60%, the trained prediction model can be compared with the model trained with the same number of real samples which achieves the same prediction effect. This further proves the effectiveness of augmenting the data set by CWGAN. When the proportion of generated data is higher than 60%, the MAE and RMSE of DNN, LSTM, and AdCNN all have a certain upward trend, which indicates that when the percentage of generated data is relatively large, the predictive ability of all models will decrease. By contrast, AdCNN can still maintain a low prediction error of RUL. In CWGAN generation experiments with different proportions, the MSE predicted by AdCNN URL decreased by 72.16%, 52.11%, and 32.96% compared with the other three comparison methods, respectively. From the overall performance, the RUL prediction method combined with CWGAN and AdCNN can effectively alleviate the problem of small samples, which has the lowest error. In addition, compared with SVR, the prediction effects of the three methods are still better, which further proves that the deep learning approach is better than the machine study approach and has strong robustness and generalization ability. In summary, when there are few available training samples, we can use CWGAN to generate part of the training data and mix it into the original sample to train AdCNN model to realize similar prediction results as the training model with the same number of real samples.

5. Results and Discussion

In real-world industrial environments, deep learning models based on data-driven for RUL prediction are often hampered by few samples. In this work, a novel prognostic framework comprising conditional Wasserstein distance-based generative adversarial networks (CWGAN) and adversarial

convolution neural networks (AdCNN) is proposed. The CWGAN model uses generators and discriminators for continuous adversarial training with Wasserstein distance. After the network achieves Nash equilibrium, the overall distribution of the CWGAN-generated data is basically consistent with the original frequency domain data, and numbers of high-quality CWGAN-generated training data can be obtained stably from the generator, so as to realize the augmentation of the data to solve the problem of few samples. In AdCNN, adversarial training is adopted for training the prediction model; that is, the predictor uses a one-dimensional convolutional neural network to extract the frequency domain features of the bearing layer by layer and then outputs the predicted value. The discriminator calculates the mean square error and judges the prediction effect of the predictor to promote adversarial training of the model. Batch normalization and dropout techniques are applied to accelerate adversarial model training, overcome vanishing gradient problems, and avoid overfitting. Then, the exponential smoothing method is used to solve the local volatility problem of bearing RUL prediction. Via the PHM 2012 challenge datasets, the performance of the proposed method is verified and compared to the methods based on SVR, DNN, and LSTM. The comparison results show that the RUL prediction method combined with CWGAN and AdCNN has more advantages, which can not only effectively alleviate the problem of few samples but also have high accuracy of bearing remaining useful life prediction.

Recently, many researchers have conducted massive studies on few-shot learning, among which meta-learning has achieved good results in few-shot learning. In view of the fact that it is not simple to obtain massive bearing degradation information, our team will use meta-learning to realize bearing RUL later on.

Data Availability

The experimental data used the bearing degradation data set of the IEEE 2012 PHM Data Challenge. <https://github.com/wkzsl111/phm-ieee-2012-data-challenge-dataset>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Our research was sponsored by the China Natural Science Foundation (No. 61871432) and the Natural Science Foundation of Hunan Province (nos. 2020JJ4275 and 2021JJ50049), and Hunan High-Tech Industry Innovation Leading Plan Project (no. 2021GK4008).

References

- [1] B. Wu, W. Li, and M. Q. Qiu, "Remaining useful life prediction of bearing with vibration signals based on a novel indicator," *Shock and Vibration*, vol. 2017, pp. 1–10, 2017.
- [2] B. Wang, Y. Lei, N. Li, and T. Yan, "Deep separable convolutional network for remaining useful life prediction of machinery," *Mechanical Systems and Signal Processing*, vol. 134, Article ID 106330, 2019.
- [3] S. Yin, X. Li, and H. Gao et al., "Data-based techniques focused on modern industry: an overview," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 657–667, 2014.
- [4] Y. Lei, N. Li, G. Li, T. Yan, and J. Lin, "Machinery health prognostics: a systematic review from data acquisition to RUL prediction," *Mechanical Systems and Signal Processing*, vol. 104, pp. 799–834, 2018.
- [5] R. Yan, R. X. Gao, and X. Chen, "Wavelets for fault diagnosis of rotary machines: a review with applications," *Signal Processing*, vol. 96, pp. 1–15, 2014.
- [6] J. Cui, L. Ren, X. Wang, and L. Zhang, "Pairwise comparison learning based bearing health quantitative modeling and its application in service life prediction," *Future Generation Computer Systems*, vol. 97, pp. 578–586, 2019.
- [7] H. Cui, Y. Qiao, Y. Yin, and M. Hong, "An investigation of rolling bearing early diagnosis based on high-frequency characteristics and self-adaptive wavelet de-noising," *Neurocomputing*, vol. 216, pp. 649–656, 2016.
- [8] J. A. A. Daviu, M. R. Guasp, J. P. Llinares, R. J. Folch, R. B. Perez, and C. C. Perez, "Toward condition monitoring of damper windings in synchronous motors via EMD analysis," *IEEE Transactions on Energy Conversion*, vol. 27, no. 2, pp. 432–439, 2012.
- [9] C. U. Mba, V. Makis, S. Marchesiello, A. Fasana, and L. Garibaldi, "Condition monitoring and state classification of gearboxes using stochastic resonance and hidden Markov models," *Measurement*, vol. 126, pp. 76–95, 2018.
- [10] F. Wang, X. Chen, B. Dun, D. Yan, and H. Zhu, "Rolling Bearing Reliability Assessment via Kernel Principal Component Analysis and Weibull Proportional hazard Model," *Shock and Vibration*, vol. 2017, Article ID 6184190, 2017.
- [11] Y. Miao, M. Zhao, J. Lin, and Y. Lei, "Application of an improved maximum correlated kurtosis deconvolution method for fault diagnosis of rolling element bearings," *Mechanical Systems and Signal Processing*, vol. 92, pp. 173–195, 2017.
- [12] C. Li, Z. Xiao, X. Xia, W. Zou, and C. Zhang, "A hybrid model based on synchronous optimisation for multi-step short-term wind speed forecasting," *Applied Energy*, vol. 215, pp. 131–144, 2018.
- [13] O. C. Reyes, P. Scully, and K. B. Ozanyan, "Deep neural networks for learning spatio-temporal features from tomography sensors," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 1, pp. 645–653, 2017.
- [14] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213–237, 2019.
- [15] X. Li, W. Zhang, and Q. Ding, "Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction," *Reliability Engineering & System Safety*, vol. 182, pp. 208–218, 2019.
- [16] J. R. Jiang, J. E. Lee, and Y. M. Zeng, "Time series multiple channel convolutional neural network with attention-based long short-term memory for predicting bearing remaining useful life," *Sensors*, vol. 20, no. 1, p. 166, 2020.
- [17] Y. Qin, D. Chen, and S. Xiang, C. Zhu, Gated dual attention unit neural networks for remaining useful life prediction of rolling bearings," *IEEE Transactions on Industrial Informatics*, vol. 17, pp. 6438–6447, 2020.
- [18] D. Verstraete, E. Droguett, and M. Modarres, "A deep adversarial approach based on multi-sensor fusion for semi-supervised remaining useful life prognostics," *Sensors*, vol. 20, no. 1, p. 176, 2020.
- [19] A. L. Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, and H. Zhang, "Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture," *Reliability Engineering & System Safety*, vol. 183, pp. 240–251, 2019.
- [20] J. Zhu, N. Chen, and W. Peng, "Estimation of bearing remaining useful life based on multiscale convolutional neural network," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 3208–3216, 2018.
- [21] W. Mao, Y. Liu, L. Ding, and Y. Li, "Imbalanced fault diagnosis of rolling bearing based on generative adversarial network: a comparative study," *IEEE Access*, vol. 7, pp. 9515–9530, 2019.
- [22] S. Xiang, Y. Qin, C. Zhu, Y. Wang, and H. Chen, "Long short-term memory neural network with weight amplification and its application into gear remaining useful life prediction," *Engineering Applications of Artificial Intelligence*, vol. 91, Article ID 103587, 2020.
- [23] S. Xiang, Y. Qin, J. Luo, H. Pu, and B. Tang, "Multicellular LSTM-based deep learning model for aero-engine remaining useful life prediction," *Reliability Engineering & System Safety*, vol. 216, Article ID 107927, 2021.
- [24] I. J. Goodfellow, J. P. Abadie, M. Mirza et al., "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.
- [25] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang, "Camera style adaptation for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5157–5166, Salt Lake City, UT, USA, June 2018.
- [26] D. A. B. Oliveira, A. B. Mattos, and E. D. S. Morais, "Improving Viseme Recognition Using GAN-Based Frontal View Mapping," in *Proceeding of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)/CVPR Workshops*, pp. 2148–2155, Lake City, UT, USA, June 2018.

- [27] S. W. Huang, C. T. Lin, and S. P. Chen, Y. Y. Wu, P. H. Hsu, and S. H. Lai, Auggan: cross domain adaptation with gan-based data augmentation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 718–731, Munich, Germany, September 2018.
- [28] M. Mirza and S. Osindero, “Conditional Generative Adversarial Nets,” 2014, <http://arxiv.org/abs/1411.1784>.
- [29] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *Proceedings of the International conference on machine learning, PMLR*, vol. 214-223, Sydney, Australia, August 2017.
- [30] I. Gulrajani, F. Ahmed, and M. Arjovsky, V. Dumoulin and A. Courville, Improved Training of Wasserstein gans,” 2017, <http://arxiv.org/abs/1704.00028>.
- [31] B. Devon, K. Nikolaos, and S. Rickard, J. Niklewski, Automatic robust estimation for exponential smoothing: perspectives from statistics and machine learning,” *Expert Systems with Applications*, vol. 160, Article ID 113637, 2020.
- [32] L. Wen, X. Li, and L. Gao, 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, 2017.
- [33] R. K. Singleton, E. G. Strangas, and S. Aviyente, “Time–frequency complexity based remaining useful life (RUL) estimation for bearing faults,” in *Proceedings of the IEEE international symposium on diagnostics for electric machines, power electronics and drives*, pp. 600–606, Valencia, Spain, August 2013.
- [34] P. Nectoux, R. Gouriveau, K. Medjaher et al., “PRONOSTIA: an experimental platform for bearings accelerated degradation tests,” in *Proceedings of the IEEE International Conference on Prognostics and Health Management*, vol. 1-8, Detroit (Romulus), MI, USA, June 2012.
- [35] R. K. Singleton, E. G. Strangas, and S. Aviyente, “Extended Kalman filtering for remaining-useful-life estimation of bearings,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1781–1790, 2014.
- [36] A. Soualhi, K. Medjaher, and N. Zerhouni, “Bearing health monitoring based on Hilbert–Huang transform, support vector machine, and regression,” *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 1, pp. 52–62, 2014.
- [37] L. Ren, Y. Sun, J. Cui, and L. Zhang, “Bearing remaining useful life prediction based on deep autoencoder and deep neural networks,” *Journal of Manufacturing Systems*, vol. 48, pp. 71–77, 2018.
- [38] F. Wang, X. Liu, G. Deng, X. Yu, H. Li, and Q. Han, “Remaining life prediction method for rolling bearing based on the long short-term memory network,” *Neural Processing Letters*, vol. 50, no. 3, pp. 2437–2454, 2019.
- [39] Y. Lei, *Intelligent Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery*, Butterworth-Heinemann, Oxford, England, 2016.