IOP Publishing Measurement Science and Technology
Meas. Sci. Technol. 31 (2020) 045006 (26pp) https://doi.org/10.1088/1361-6501/ab47df

# A Wasserstein gradient-penalty generative adversarial network with deep auto-encoder for bearing intelligent fault diagnosis

**Xiong Xiong, Jiang Hongkai[1], Xingqiu Li and Maogui Niu**

School of Aeronautics, Northwestern Polytechnical University, 710072, Xi'an, People's Republic of China

E-mail: jianghk@nwpu.edu.cn

Received 28 June 2019, revised 15 September 2019
Accepted for publication 25 September 2019
Published 9 January 2020

**Abstract**

It is a great challenge to manipulate unbalanced fault data in the field of rolling bearings intelligent fault diagnosis. In this paper, a novel intelligent fault diagnosis method called the Wasserstein gradient-penalty generative adversarial network with deep auto-encoder is proposed for intelligent fault diagnosis of rolling bearings. Firstly, the gradient penalty term is added to the Wasserstein generative adversarial network to enhance the stability and convergence of the network. Secondly, a deep auto-encoder network comprised of multiple auto-encoders is regarded as the discriminator. Finally, the sparse auto-encoder is placed at the end of the proposed method as the classifier to classify synthetic bearing faults. The results show that the proposed method has a better performance than traditional methods and the Wasserstein generative adversarial network.

Keywords: rolling bearing, intelligent fault diagnosis, Wasserstein gradient-penalty generative adversarial network, deep auto-encoder, unsupervised learning

(Some figures may appear in colour only in the online journal)

## 1. Introduction

Rotating machines have become more and more widely used in modern life and industry. With the rapid development of science and technology, the latest rotating machinery has become more high speed and more integrated. The rolling bearing, one of the most critical components in the rotating machine, will inevitably fail after a long period of operation or in a terrible working environment, which will lead to unnecessary safety accidents and economic losses. Therefore, accurate and reliable fault diagnosis of rolling bearings is always of great practical significance.

In the field of rolling bearing fault diagnosis, as well as the traditional signal processing methods, more and more intelligent diagnosis methods are now being applied [1, 2]. Various methods based on machine learning [3] are at the forefront, especially the most popular deep learning [4, 5], such as convolutional neural networks (CNN) [6, 7], recurrent neural networks (RNN) [8, 9], and deep belief networks (DBN) [10]. The auto-encoder (AE) is also one of these – an unsupervised deep learning network [11, 12]. An AE is a special neural network that consists of three layers: an input layer, a hidden layer, and an output layer. In the structure of an AE, the input and output layers have the same number of neurons [13]. The structure of an AE can be considered as an encoder which is integrated with a decoder. The encoder includes the input layer and the hidden layer, mapping the input vector to the hidden layer. The decoder takes the output of the hidden layer to recreate the input values [14]. Generally, the dataset used to train a neural network needs to be very large and balanced. It will be difficult for a neural network to train on seriously unbalanced data, and the generalization ability is not strong. Furthermore, it requires a lot of time, material resources, and human resources to collect the mass fault data of the rolling bearing equipment. However, the amount of existing fault data is basically quite small. In this situation, a larger amount of fault data is urgently needed.

An unsupervised generation model named the generative adversarial network (GAN) was proposed by Goodfellow

[1] Author to whom any correspondence should be addressed.

 

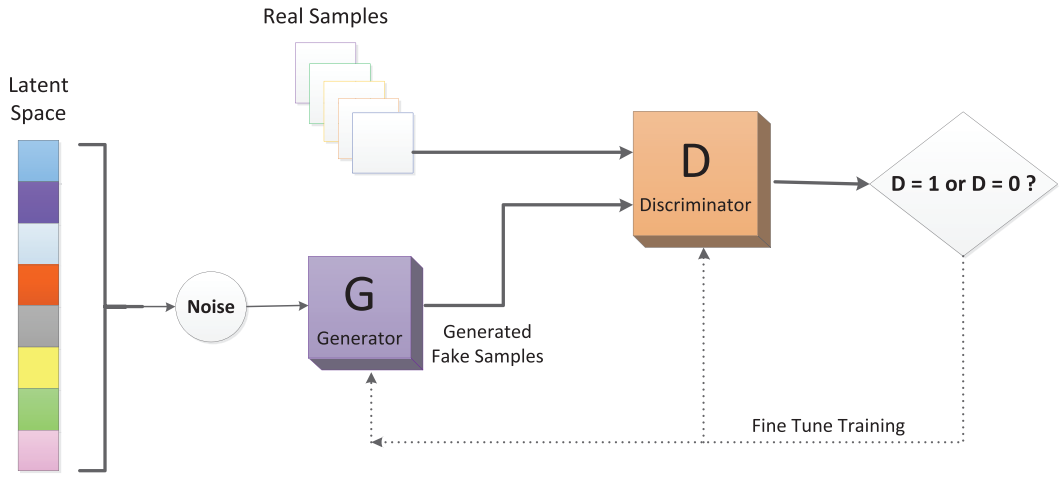


**Figure 1.** Training structure of GAN.

*et al* [15] at the 2014 Neural Information Processing Systems (NIPS) conference. It can artificially generate samples of fitting real data from random noise samples according to real sample distribution [16, 17]. For a deep Boltzmann machine, GAN does not need to repeatedly apply a Markov chain to generate [18]. Our team is mainly doing research on rolling bearing fault diagnosis [19], where the amount of fault data required is very large, so we intend to make use of GAN to do research on data generation.

However, during the training process of GAN, it has a gradient vanishing problem [19]. In order to solve the problem, in 2017, an improved model named the Wasserstein generative adversarial network (WGAN) was proposed by Martin *et al* [20], which replaces Jensen–Shannon divergence with Wasserstein distance. It has already been applied to image [21, 22].

In WGAN, the weight needs to be limited to a range to satisfy Lipschitz continuity, but it easily leads to a gradient explosion problem. The weight clipping in WGAN will lead to optimization difficulties, and the results will have a pathological value surface. In order to solve this problem, this paper proposes to use the gradient penalty methods to satisfy the continuity condition.

In this paper, a novel deep unsupervised learning method called the Wasserstein gradient-penalty generative adversarial network (WGGAN) with deep auto-encoder (DAE) is proposed for rolling bearing intelligent fault diagnosis. Firstly, the gradient penalty term is added to the WGAN to enhance the stability and convergence of the network. Secondly, a deep auto-encoder network comprised of multiple auto-encoders is regarded as the discriminator. Finally, the sparse auto-encoder is placed at the end of the proposed method as the classifier to classify synthetic bearing faults. The results show that the proposed method can get rid of gradient vanishing and gradient explosion, and the diagnostic accuracy of the proposed method is higher than that of other methods.

The rest of this paper is organized as follows. The theory of standard WGAN is briefly introduced in section 2. The proposed method is described in detail in section 3. The results of the verification experiment are analyzed and discussed in section 4. The results of the engineering application experiment are analyzed and discussed in section 5. Finally, a general conclusion is given in section 6.

## 2. Standard theory of WGAN and auto-encoder

### *2.1. Description of recent intelligent fault diagnosis progress*

As artificial intelligence methods have rapidly developed, a large number of intelligent fault diagnosis papers have been published. In 2018, a novel method called deep wavelet auto-encoder (DWAE) with extreme learning machine (ELM) was proposed by Shao *et al* [4] for intelligent fault diagnosis of rolling bearings. Wang *et al* [23] adopted a convolutional neural network-based hidden Markov model (CNN-HMM) to classify multifaults in mechanical systems. Jiao *et al* [24] presented a multivariate encoder information-based convolutional neural network (MEI-CNN) for intelligent diagnosis. In 2019, Yan *et al* [25] studied a novel fault diagnosis technique based on improved multiscale dispersion entropy (IMDE) and max-relevance min-redundancy (mRMR) to efficiently extract fault feature information and improve fault diagnosis accuracy. Wang *et al* [26] introduced a new feature learning method for fault diagnosis of planetary gearboxes based on deep conditional variational neural networks (CVNN). Wang *et al* [27] proposed a recently developed optimization method called batch normalization into deep neural networks (DNN) to realize online monitoring and fast fault diagnosis.

### *2.2. Standard GAN method*

The main inspiration of GAN originates from the idea of a two-person zero-sum game in game theory, which sets two participating players as a generator and a discriminator respectively. The purpose of the generator is to learn and capture the potential distribution of real data samples as much as possible. The discriminator is a two-classifier, the purpose of which is to correctly distinguish whether the input data are real data or from the generator. The whole learning optimization process

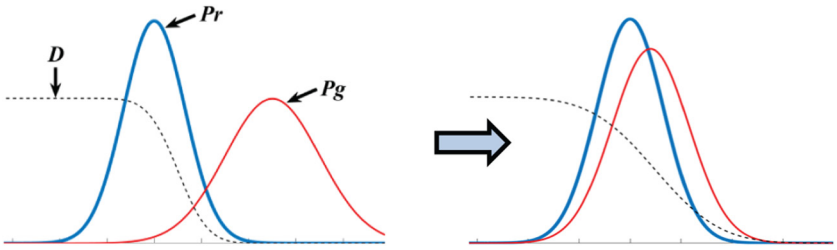


**Figure 2.** The sample trend of GAN.

is a minimax game problem, the ultimate purpose of which is to find a Nash equilibrium between the two, letting the generator estimate the distribution of the data samples. The training structure of GAN is shown in figure 1.

The optimization problem of GAN is a minimax problem. The objective function of GAN can be described as follows:

$$\min_G \max_D V(D,G) = E_{x\sim p_r(x)}[\log D(x)] + E_{y\sim p_y(y)}[\log(1-D(G(y)))]. \quad (1)$$

The generator $G$ defines implicitly the probability distribution $p_g$ as the distribution of the sample $G(y)$ obtained when $y \sim p_y$ in equation (1). Therefore, if sufficient capacity and training time are given, it is hoped that GAN will converge to a good $p_r$ estimator.

When the generator $G$ is fixed, the best discriminator $D$ is:

$$D_G^*(x) = \frac{p_r(x)}{p_r(x) + p_g(x)}. \quad (2)$$

As shown in figure 2, the solid blue line represents the distribution of the raw data samples $p_r$, the solid red line represents the distribution of the generated sample $p_g$, and the dotted black line represents the constantly updated distribution of the discriminator. When the generating network is trained, the discriminator $D$ is updated at the same time so that $D$ can distinguish samples in $p_r$ and samples in $p_g$.

### 2.3. Standard WGAN method

There exist two forms of problem in the original standard GAN: (1) the better the discriminator is, the more seriously the generator gradient vanishes; (2) minimizing the loss function of the second generator is equivalent to minimizing an unreasonable distance measure, which leads to gradient instability and collapse mode (insufficient diversity).

The nature of WGAN is to replace Jensen–Shannon divergence with Wasserstein distance. Wasserstein distance is also called Earth-mover (EM) distance [28], defined as follows:

$$W(P_r,P_g) = \frac{1}{N}\sup_{\|f\|_L\leqslant N} E_{x\sim P_r}[f(x)] - E_{x\sim P_g}[f(x)]. \quad (3)$$

If a set of parameters $\theta$ is used to define a series of possible functions $f_\theta$, then equation (3) can be approximated as follows:

$$N\cdot W(P_r,P_g) \approx \max_{\theta:|f_\theta|_L\leqslant N} E_{x\sim P_r}[f_\theta(x)] - E_{x\sim P_g}[f_\theta(x)] \quad (4)$$

where $f$ can be represented by a neural network with a parameter $\theta$. Owing to the fact that the neural network is strong enough to fit, the series of $f_\theta$ defined in this way is enough to approximate highly $\sup_{f_L\leqslant N}$ in equation (3).

At this point, a discriminator network $f_\theta$ with parameter $\theta$, and in which the last layer is a linear activation layer, can be constructed, under the condition where $\theta$ does not exceed a certain range, make $L$ in equation (5) as large as possible:

$$L = E_{x\sim P_r}[f_\theta(x)] - E_{x\sim P_g}[f_\theta(x)]. \quad (5)$$

where $L$ will approximate the Wasserstein distance between real distribution and generated distribution [29].

There are two loss functions of WGAN: the loss function of the generator is equation (6); the loss function of the discriminator is equation (7).

$$-E_{x\sim P_g}[f_\theta(x)] \quad (6)$$

$$E_{x\sim P_g}[f_\theta(x)] - E_{x\sim P_r}[f_\theta(x)]. \quad (7)$$

### 2.4. Standard method of an auto-encoder

An auto-encoder is a multilayer neural network in which the input layer and the output layer represent the same meaning, with the same number of nodes. An identity function with the same input and output is learned by the auto-encoder. The purpose of an auto-encoder is to learn the middle code layer (usually the layer with fewer nodes, or the middlemost layer), which is a good representation of the input vector. This process has played a role in dimensionality reduction. When the auto-encoder has only one hidden layer, its principle is equivalent to principal component analysis (PCA). When the auto-encoder has multiple hidden layers, a restricted Boltzmann machine (RBM) can be used to pre-train between each two layers, and a backpropagation algorithm is used to adjust the final weight. The weight updating of the network is deduced by calculating partial derivative, and the algorithm is a gradient descent method.

An auto-encoder is an unsupervised learning algorithm that uses the backpropagation algorithm to make the target value approximately equal to the input value. An auto-encoder tries to learn a function $f_{a,b}(\mathrm{x}) \approx x$, which means the auto-encoder tries to approximate an identity function. The network structure of the auto-encoder is shown in figure 3. The auto-encoder can learn some compressed representations of

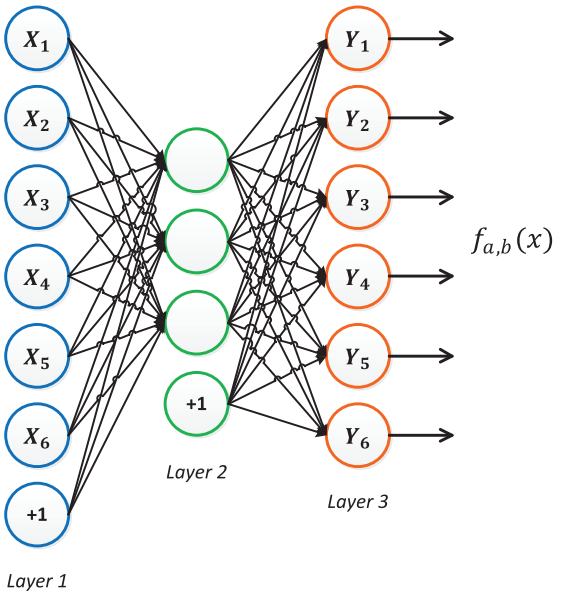


**Figure 3.** The network structure of an auto-encoder.

data, so an auto-encoder is a way to learn the correlation of input data.

The purpose of the backpropagation algorithm is to find the minimum value of function $J(a,b)$ for $a$ and $b$. Firstly, each parameter $W_{ij}^{(l)}$ and $b_i^{(l)}$ needs to be initialized to a very small random value close to 0, and then the weight is updated iteratively by a gradient descent method.

$$J(a,b;x,y) = \frac{1}{2}\|f_{a,b}(x) - y\|^2. \tag{8}$$

Equation (8) is the cost function of a single $(x,y)$ for the sample set $\{(x^1,y^1),(x^2,y^2),\ldots,(x^m,y^m)\}$.

$$J(a,b) = \left[\frac{1}{m}\sum_{i=1}^{m} J(a,b;x,y)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_{l-1}}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}\left(W_{ij}^{(l)}\right)^2. \tag{9}$$

Equation (9) is the cost function of the whole sample set. The first term is the mean square deviation and the second term is regularization. In order to prevent over-fitting, $\lambda$ is used to control the correlation between the two terms.

$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha\frac{\partial}{\partial W_{ij}^{(l)}}J(a,b) \tag{10}$$

$$b_i^{(l)} = b_i^{(l)} - \alpha\frac{\partial}{\partial b_i^{(l)}}J(a,b) \tag{11}$$

where $\alpha$ is the learning rate and is an important parameter. The biggest problem here is to find partial derivatives.

## 3. The proposed method

### 3.1. WGGAN design

Firstly, let $P_r$ and $P_g$ be two distributions in X, which is a compact metric space. Then there is a 1-Lipschtiz function $f^*$, which is the optimal solution of $\max_{\|f\|_L\leqslant 1} E_{y\sim P_r}[f(y)] - E_{x\sim P_g}[f(x)]$. Let $\pi$ be the optimal between $P_r$ and $P_g$, defined as the minimizer of $W(P_r,P_g) = \inf_{\pi\in\prod(P_r,P_g)} E_{(x,y)\sim\pi}[\|x-y\|]$. $\prod(P_r,P_g)$ is the set of joint distributions $\pi(x,y)$ whose marginals are $P_r$ and $P_g$ respectively. Then, if $f^*$ is differentiable, $\pi(x=y)=0$ and $x_t = tx + (1-t)y$ with $0\leqslant t\leqslant 1$, it holds that $P_{(x,y)\sim\pi}\left[\nabla f^*(x_t) = \frac{y-x_t}{\|y-x_t\|}\right] = 1$. So, $f^*$ has gradient norm 1 almost everywhere under $P_r$ and $P_g$.

**Table 1.** The construction of WGGAN.

Algorithm. The training process of a Wasserstein generative adversarial network with gradient penalty.

**Input:**

learning rate $\alpha$; gradient penalty coefficient $\lambda$; batch size $N$; the discriminator's updating number $K$

**while** $\varphi$ has not converged do

  **for** $k = 1$ to $K$ do

    **for** $j = 1$ to $N$ do

    Sample a pair of true samples $x\sim P_r$ and noise samples $z\sim P_z$, and a random number $\beta\sim U[0,1]$

    $y = G(z)$

    $z = \beta x + (1-\beta)y$

    $L_j = D_\theta(y) - D_\theta(x) + \lambda(\nabla_z D_\theta(z)_2 - 1)^2$

    **end for**

    $\theta = \mathrm{Adam}\left(\nabla_\theta \frac{1}{N}\sum_{j=1}^{N} L_j, \theta, \gamma_1, \gamma_2\right)$.

  **end for**

  Sample batch of N noise samples $\{z_1,\ldots,z_n\}$ from $P_z$

  $\varphi = \mathrm{Adam}\left(\nabla_\varphi \frac{1}{N}\sum_{j=1}^{N} -D_\theta(G(z)), \varphi, \gamma_1, \gamma_2\right)$

**end while**

It is observed that the WGAN optimization process is difficult because of interactions between the weight constraint and the cost function, so it leads to vanishing or exploding gradients without careful tuning of the clipping threshold.

Now, gradient penalty is proposed to establish a loss function to satisfy the Lipschitz limit that requires the gradient of the discriminator not to exceed K. Firstly, the gradient $d(D(x))$ of the discriminator is found, and then a two-norm is established between it and K to achieve a simple loss function. Focusing on the generated sample concentrated area, the real sample concentrated area, and the area sandwiched between them. In order to avoid more issues, the soft version of the constraint is enforced with a penalty on the gradient norm for random samples $z\sim P_z$. The new objective will be changed.

Firstly, a pair of true and fake samples is randomly sampled, and a random number from zero to one:

$$x\sim P_r, \quad y\sim P_g, \quad \beta\sim \mathrm{Uniform}[0,1]. \tag{12}$$

Then, random sampling with interpolation on the line between x and y:

$$z = \beta x + (1-\beta)y. \tag{13}$$

The distribution satisfied by z sampled according to the above process is denoted as $P_z$, and the final version of the discriminator loss function is obtained:

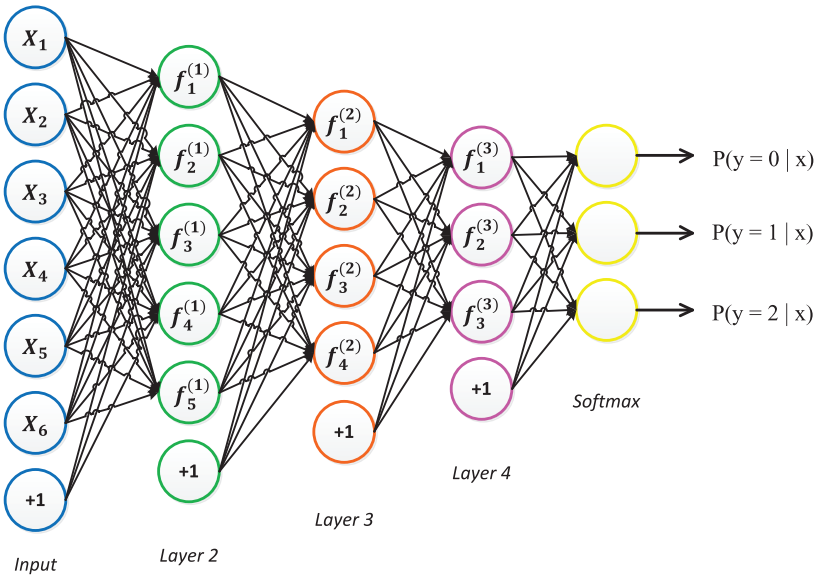


**Figure 4.** The network structure of a deep auto-encoder.

$$L(D) = \underbrace{E_{y\sim P_g}[D(y)] - E_{x\sim P_r}[D(x)]}_{\text{Original critic loss}} + \underbrace{\lambda E_{z\sim P_z}\left[\left(\|\nabla_z D(z)\|_2 - 1\right)^2\right]}_{\text{The gradient penalty}}. \tag{14}$$

Equation (14) is the new objective after the gradient penalty is added to the proposed method. The improved new generation model is called WGGAN, the algorithm for which is shown in table 1.

(1) The weight clipping method is valid for the global sample space, but because the gradient norm of the discriminator is indirectly limited, it will lead to the gradient vanishing or gradient explosion.
(2) The gradient penalty method only takes effect on the true and false sample concentrated area and the transition zone in the middle, but in this way the gradient controllability is very strong and it is easy to adjust to the appropriate scale because the gradient of the discriminator is directly limited to one.

Owing to the fact that the gradient penalty is applied independently to each sample, the batch normalization cannot be used in the model structure of the discriminator. It will introduce the interdependencies of different samples in the same batch. Other normalization methods can be selected. The normalization method used in the paper is layer normalization [30].

### *3.2. The construction of a deep auto-encoder*

It is easy to converge to a local minimum using the backpropagation algorithm, but it is not possible to get a good classification result. To solve this problem, a layer-by-layer greedy algorithm can be used to train the deep network, as shown in figure 4. First, the original input is used to train the first layer of the network, to obtain parameters $a^{(1,1)}, b^{(1,1)}, a^{(1,2)}, b^{(1,2)}$. Then the first layer of the network transforms the original input into a vector A, consisting of the activation values of hidden units, then takes A as the input of the second layer and continues training to get parameters of the second layer $a^{(2,1)}, b^{(2,1)}, a^{(2,2)}, b^{(2,2)}$. Finally, the same strategy is adopted for the following layers: the output of the former layer is trained as the input for the next layer in turn. When training the parameters of each layer, the parameters of the other layers will be fixed and kept unchanged. In order to get better results, after the above training process is completed, the parameters of all layers can be adjusted simultaneously by the backpropagation algorithm. This process is commonly referred to as fine-tuning.

The process of fine-tuning is shown as follows:

(1) Activation A of each hidden layer is calculated by a forward propagation algorithm.
(2) For the output layer $L_{n_l}$, equation (15) is calculated.

$$\delta^{n_l} = -(y - a^{n_l}) \cdot f'(z^{n_l}) \tag{15}$$

(3) For every layer of $l = n_l - 1, n_l - 2, \ldots, 2$, equation (16) is calculated.

$$\delta^{n_l} = \left(\left(a^{(l)}\delta^{(l+1)}\right)\right) \cdot f'(z^l). \tag{16}$$

(4) The final partial derivative value is calculated as follows:

$$\nabla_{a^l} J(a, b; x, y) = \delta^{(l+1)} \cdot \left(a^{(l)}\right)^T \tag{17}$$

$$\nabla_{b^l} J(a, b; x, y) = \delta^{(l+1)}. \tag{18}$$

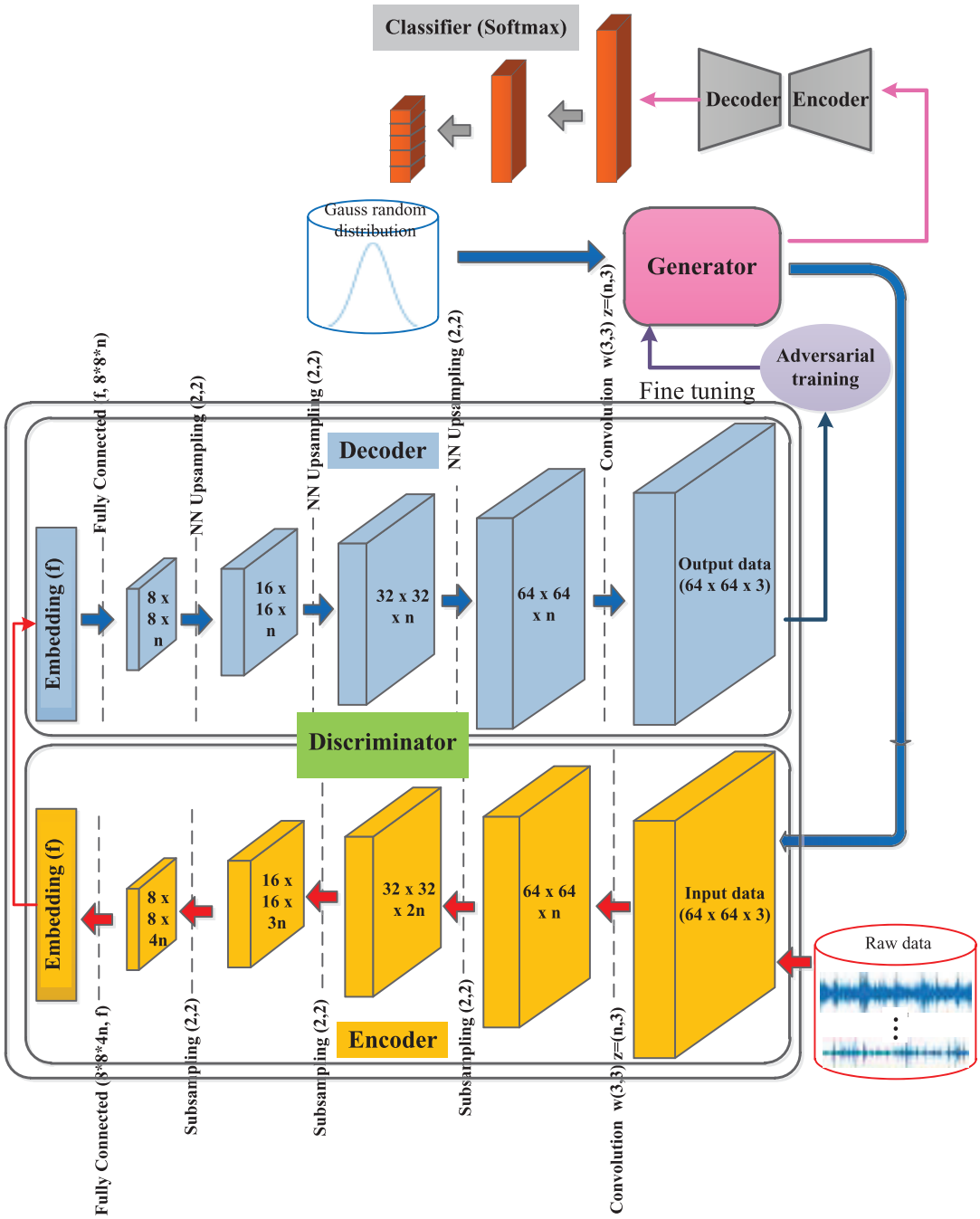


**Figure 5.** The construction of WGGAN with deep auto-encoder.

### *3.3. The construction of the proposed method*

In this paper, a novel method called the Wasserstein gradient-penalty generative adversarial network with DAE is proposed to expand the amount of fault data. In the WGGAN method, the generator network and the discriminator network are two independent network structures. The result generated by the generator is regarded as the input of the discriminator. Then, the result of calculating the error function in the discriminator is fed back to the generator to fine-tune the network parameters, generating new results again. This is a constant cycle. In the paper, the deep auto-encoder is proposed to act as the discriminator network to discriminate the input, as shown in figure 5.

The proposed method aims to use a loss derived from the Wasserstein distance to match auto-encoder loss distributions. A typical GAN objective with the addition of an equilibrium term is used to balance the discriminator and the generator. This new method has an easier training procedure and simpler neural network architecture. Firstly, we introduce the auto-encoder loss and compute a lower bound to the Wasserstein

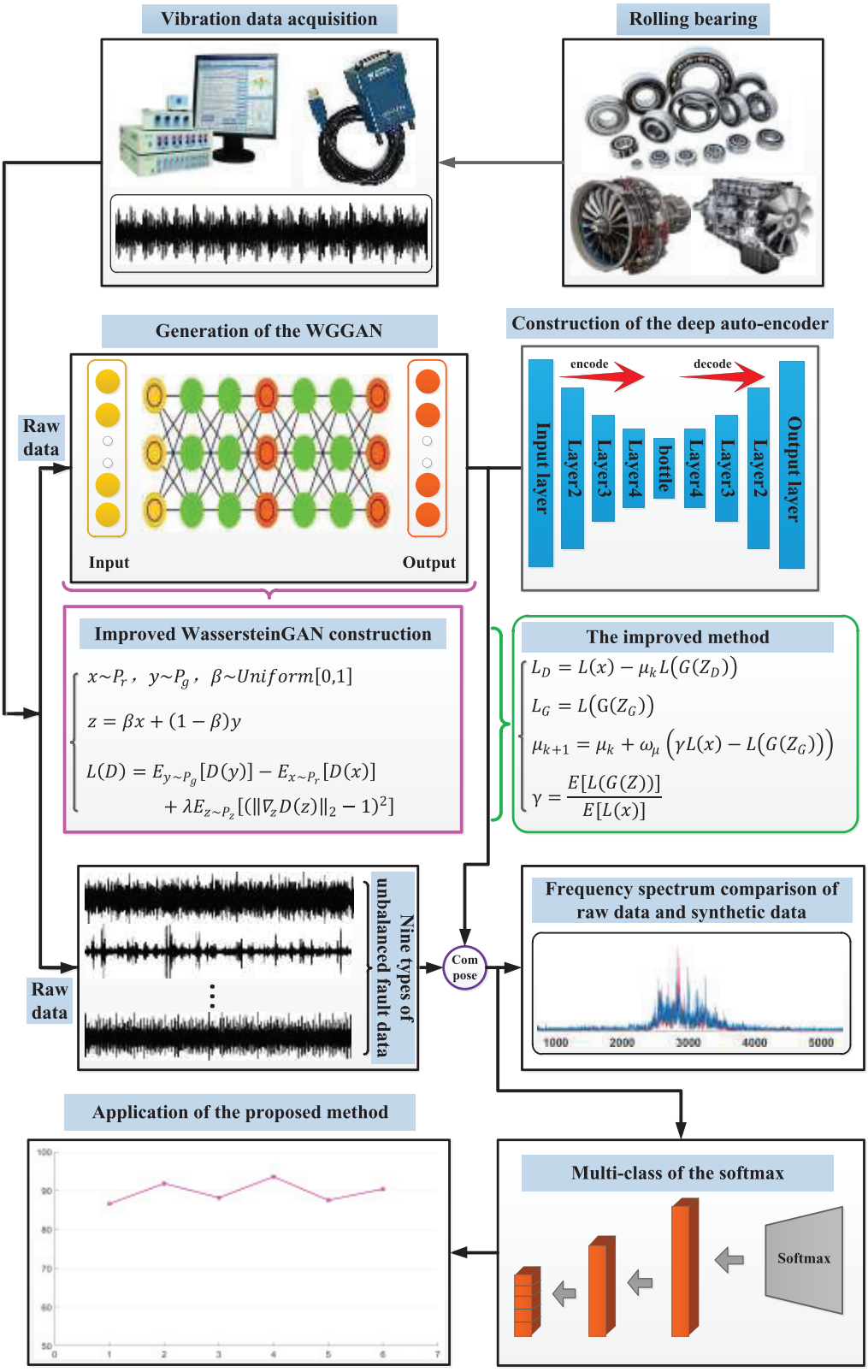


**Figure 6.** The flowchart of the proposed method.

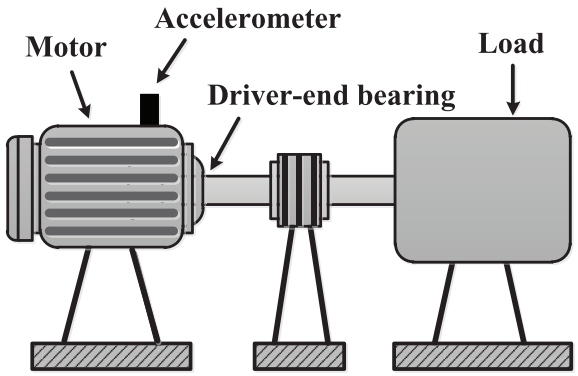


**Figure 7.** The rolling bearing experimental device.

distance between the auto-encoder loss distributions of the real and generated samples.

Secondly, $\mathcal{L}: R^{N_x} \mapsto R^{+}$ is the loss for training an auto-encoder:

$$\mathcal{L}(v) = |v - D(v)|^{\eta} \text{ where } \begin{cases} D: R^{N_x} \mapsto R^{N_x} & \text{the autoencoder function} \\ \eta \in \{1,2\} & \text{the target norm} \\ v \in R^{N_x} & \text{the sample of dimension } N_x \end{cases}. \tag{19}$$

Let $\mu_{1,2}$ be two distributions of auto-encoder losses, let $\Gamma(\mu_1, \mu_2)$ be the set of all couplings of $\mu_1$ and $\mu_2$, and let $m_{1,2} \in \mathrm{R}$ be their respective means. The Wasserstein distance can be expressed as:

$$W_1(\mu_1, \mu_2) = \inf_{\gamma \in \Gamma(\mu_1, \mu_2)} E_{(x_1, x_2) \sim \gamma}[|x_1 - x_2|]. \tag{20}$$

Using Jensen's inequality, a lower bound to $W_1(\mu_1, \mu_2)$ can be found:

$$\inf E[|x_1 - x_2|] \geqslant \inf |E[x_1 - x_2]| = |m_1 - m_2|. \tag{21}$$

This new method aims to optimize a lower bound of the Wasserstein distance between auto-encoder loss distributions, not between sample distributions.

The discriminator is designed to maximize equation (21) between auto-encoder losses. Let $\mu_1$ be the distribution of the loss $\mathcal{L}(x)$, where $x$ are real samples. Let $\mu_2$ be the distribution of the loss $\mathcal{L}(G(z))$, where $G: R^{N_z} \mapsto R^{N_x}$ is the generator function and $z \in [-1,1]^{N_z}$ are uniform random samples of dimension $N_z$.

Since $m_1, m_2 \in R^{+}$ there are only two possible solutions to maximizing $|m_1 - m_2|$:

$$\text{(a)} \begin{cases} W_1(\mu_1, \mu_2) \geqslant m_1 - m_2 \\ m_1 \to \infty \\ m_2 \to 0 \end{cases} \text{ or (b)} \begin{cases} W_1(\mu_1, \mu_2) \geqslant m_2 - m_1 \\ m_1 \to 0 \\ m_2 \to \infty \end{cases}. \tag{22}$$

The solution (b) is selected because minimizing $m_1$ leads to auto-encoding the real data. The discriminator and generator parameters $\theta_D$ and $\theta_G$ are given, each updated by minimizing the losses $\mathcal{L}_D$ and $\mathcal{L}_G$. This problem is expressed as the objective of GAN, where $z_D$ and $z_G$ are samples from $z$:

$$\begin{cases} \mathcal{L}_D = L(x, \theta_D) - L(G(z_D, \theta_G), \theta_D) \text{ for } \theta_D \\ \mathcal{L}_G = -\mathcal{L}_D \text{ for } \theta_G \end{cases}. \tag{23}$$

Similar to equations (6) and (7) from WGAN, equation (23) has one important difference: the new method matches distributions between the losses, not between the samples.

During model training, if the discriminator is not very good at identifying the real data and the false data, this will cause the generator to be confident and always generate false data with poor performance. So, in order to motivate the generator to improve its own generation ability, and not to let the generator fall into a standstill state, the discriminator must be trained several times to increase its discriminative accuracy.

The whole flowchart of the proposed method is shown in figure 6 and the general procedure is summarized as follows:

**Step 1:** The vibration signal of rolling bearings is measured by sensors and data collected in the acquisition system.

**Step 2:** Nine classes of fault data are selected as experimental data in the collected data to be separately placed in the WGGAN-DAE.

**2.1:** The real data samples are placed in the discriminator $D$, and the random noise data are placed in the generator $G$.

**2.2:** The generator $G$ generates a series of samples that are assumed to be real data and put into the discriminator $D$, and the discriminator begins to judge the real sample and the generated sample.

**2.3:** The Jensen–Shannon divergence is replaced by the Wasserstein distance as a mean that measures the distance between two sample distributions.

**2.4:** The loss function of the discriminator is established in the gradient penalty method.

**2.5:** The discriminator $D$ is constantly trained to gain the maximum loss close to the Wasserstein distance.

**2.6:** The output of $D$ is transmitted to the generator $G$ by feedback, so that the generator updates the network structure and regenerates new samples.

**2.7:** The generator and discriminator are updated repeatedly to get a good generator.

**Step 3:** The vibration signals of generated samples and raw samples are compared on the time spectrum and the frequency spectrum.

**Step 4:** The accuracy of generated extended data and raw data is compared in the deep auto-encoder.

## 4. Experimental verification

### *4.1. The data description of the rolling bearing experiment*

In this paper, the experimental vibration data of rolling bearings comes from the Electrical Engineering Lab at Case Western Reserve University; the experimental equipment in the experiment is shown in figure 7. In this study, the vibration data of drive end at the sampling frequency of 12 kHz under the same load of 0 hp is collected.

The ten rolling bearing conditions are created, and listed in table 2. Sample points are 120 000 in normal data and 12 000 in each group of fault data. There is a serious imbalance

**Table 2.** Description of the rolling bearing operation conditions.

| Conditions | Fault diameter (mm) | Outer race fault orientation | Motor speed (rpm) | Sample points |
|---|---|---|---|---|
| Normal | 0 | — | 1796 | 120 000 |
| Ball | 0.1778 | — | 1796 | 12 000 |
| Ball | 0.3556 | — | 1796 | 12 000 |
| Ball | 0.7112 | — | 1797 | 12 000 |
| Inner race | 0.1778 | — | 1797 | 12 000 |
| Inner race | 0.3556 | — | 1796 | 12 000 |
| Inner race | 0.5334 | — | 1797 | 12 000 |
| Outer race | 0.1778 | @3:00 | 1797 | 12 000 |
| Outer race | 0.1778 | @12:00 | 1797 | 12 000 |
| Outer race | 0.5334 | @12:00 | 1796 | 12 000 |

between normal and fault data. The raw data samples of ten rolling bearing conditions are shown in figure 8.

### *4.2. Qualitative analysis of fault data generation samples*

In this experiment, each class of data value is normalized to [0, 1], which means the range of each class of data point is 0 to 1, but the law of data distribution does not change, just like the original. The program of the proposed method runs with tensorflow 1.8, on i7-7700k CPU, gtx1080ti GPU. In order to compare the raw data sample and the generated extended sample in more detail, 12 000 sample points are selected to compare the feature distributions in each of the two samples. For the convenience of data distribution comparison, the raw data are also normalized to [0, 1]. The time spectrum diagram of the samples generated by WGGAN-DAE and raw samples are shown in figure 9.

It can be seen that in figure 9, the feature distribution of the sample generated by WGGAN-DAE is very close to the raw sample, so the training generator model has a great effect. In order to more specifically show the fitting degree of the original time domain signal and the generated time domain signal, the root mean square error (RMSE) is used in this paper.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2}. \tag{24}$$

The RMSE is calculated in equation (24), where $x_i$ is every time value of raw data and $y_i$ is every time value of generated data.

In table 3, the labels 1–9 correspond to fault data (Ball 0.007, Ball 0.014, Ball 0.028, Inner race 0.007, Inner race 0.014, Inner race 0.021, Outer race 0.007@3:00, Outer race 0.007@12:00, Outer race 0.021@12:00) in turn. As we can see, every RMSE of WGGAN-DAE is obviously higher than WGAN.

Next, nine classes of the raw sample and the sample generated by WGGAN-DAE will be compared in the frequency spectrum diagram, at a sampling frequency of 12 kHz. As shown in figure 10, data points of the sample within a cycle are selected.

The above is the frequency spectrum diagram comparison of the raw sample and the generated sample of nine classes of fault data at a sampling frequency of 12 kHz. It can be seen that the generated sample (blue) is very close to the raw sample (red) on the frequency spectrum diagram.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i' - y_i')^2}. \tag{25}$$

RMSE is calculated in the equation (25), where $x_i'$ is every frequency amplitude of raw data and $y_i'$ is every frequency amplitude of generated data.

In table 4, the labels 1–9 correspond to fault data (Ball 0.007, Ball 0.014, Ball 0.028, Inner race 0.007, Inner race 0.014, Inner race 0.021, Outer race 0.007@3:00, Outer race 0.007@12:00, Outer race 0.021@12:00) in turn. As we can see, every RMSE of WGGAN-DAE is obviously higher than WGAN.

### *4.3. The diagnosis results and analysis of the experiment*

The diagnosis experiment is intended to use the machine learning methods to do a multiclassification experiment on the fault data. In order to ensure that the impact of the classifier itself on the accuracy comparison is minimized in the classification experiment, four different classifiers are chosen to be added into the experiment. Based on previous experience, the machine learning methods selected are the backpropagation neural network (BPNN), random forest (RF), support vector machine (SVM), and DAE.

As shown in table 5, dataset A is raw data. In A, the normal condition consists of 150 samples, and each sample contains 800 data points. The random 100 raw samples are used for training and the 50 raw samples for testing. Each fault condition consists of 15 samples, and each sample contains 800 data points. The random ten raw samples are used for training and the five raw samples for testing. There is a serious imbalance between normal and fault data in dataset A. Dataset B is synthetic data with WGAN, and dataset C is synthetic data with WGGAN-DAE. In B and C, each condition consists of 150 samples, and each sample contains 800 data points. The random 100 generated samples of each condition are used for training and the 50 raw samples for testing. In order to better analyze the results, the ten different conditions are labelled 1 to 10.

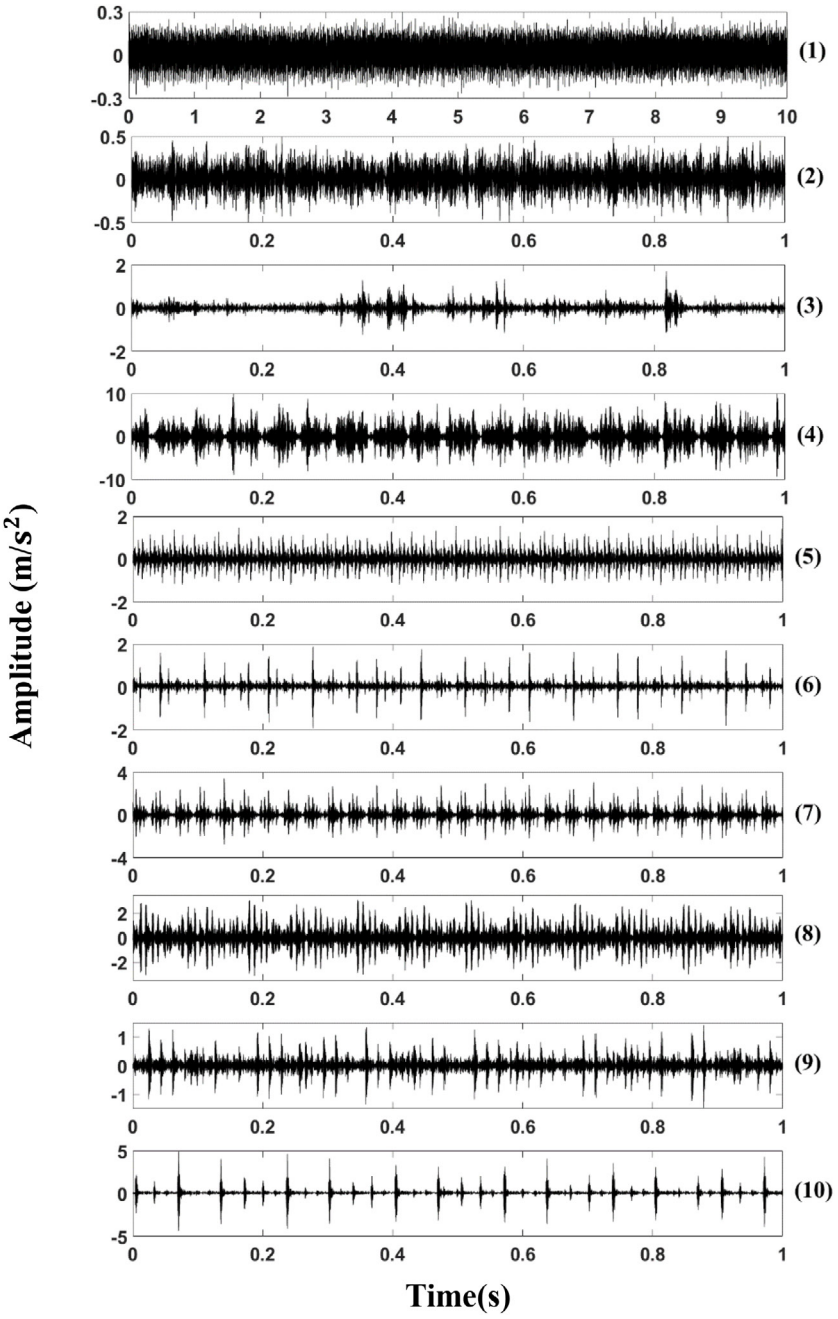


**Figure 8.** Vibration signals of the ten operating conditions: (1) normal (120 000 data points); (2) ball fault (0.007); (3) ball fault (0.014); (4) ball fault (0.028); (5) inner race fault (0.007); (6) inner race fault (0.014); (7) inner race fault (0.021); (8) outer race fault (0.007@3:00); (9) outer race fault (0.007@12:00); (10) outer race fault (0.021@12:00).

BPNN, RF, SVM, and DAE are used respectively in dataset A. WGAN is used in dataset B. WGGAN-DAE is used in dataset C. The classifiers are trained with training samples at first, and are finally tested with testing samples. In order to compare the accuracy of a little raw data and a large amount of generated data more cleanly, the data was not processed too much before the experiment, to minimize other unnecessary impacts on the results.

In order to show the stability of the proposed method, each method is tried six times. The classification testing accuracy rates of the six methods are listed in table 6, and the diagnosis results in each trial are shown in figure 11. In figure 11, the testing accuracy of the proposed method in each trial is 86.60% (433/500), 85.84% (429/500), 86.17% (430/500), 85.59% (428/500), 87.4% (437/500), 86.43% (432/500). From table 6, it is observed that the average accuracy of the proposed method is 86.60% (2598/3000), higher than BPNN, SVM, RF, and DAE using the raw data, which are 50.88% (290/570), 62.11% (354/570), 64.21 (366/570), and 68.24% (389/570) respectively. In the same amount of generated data, the diagnosis result of the proposed method is much higher than WGAN, which is 75.47% (2264/3000). So the quality of data generated by WGAN is not as good as that generated by the proposed method.

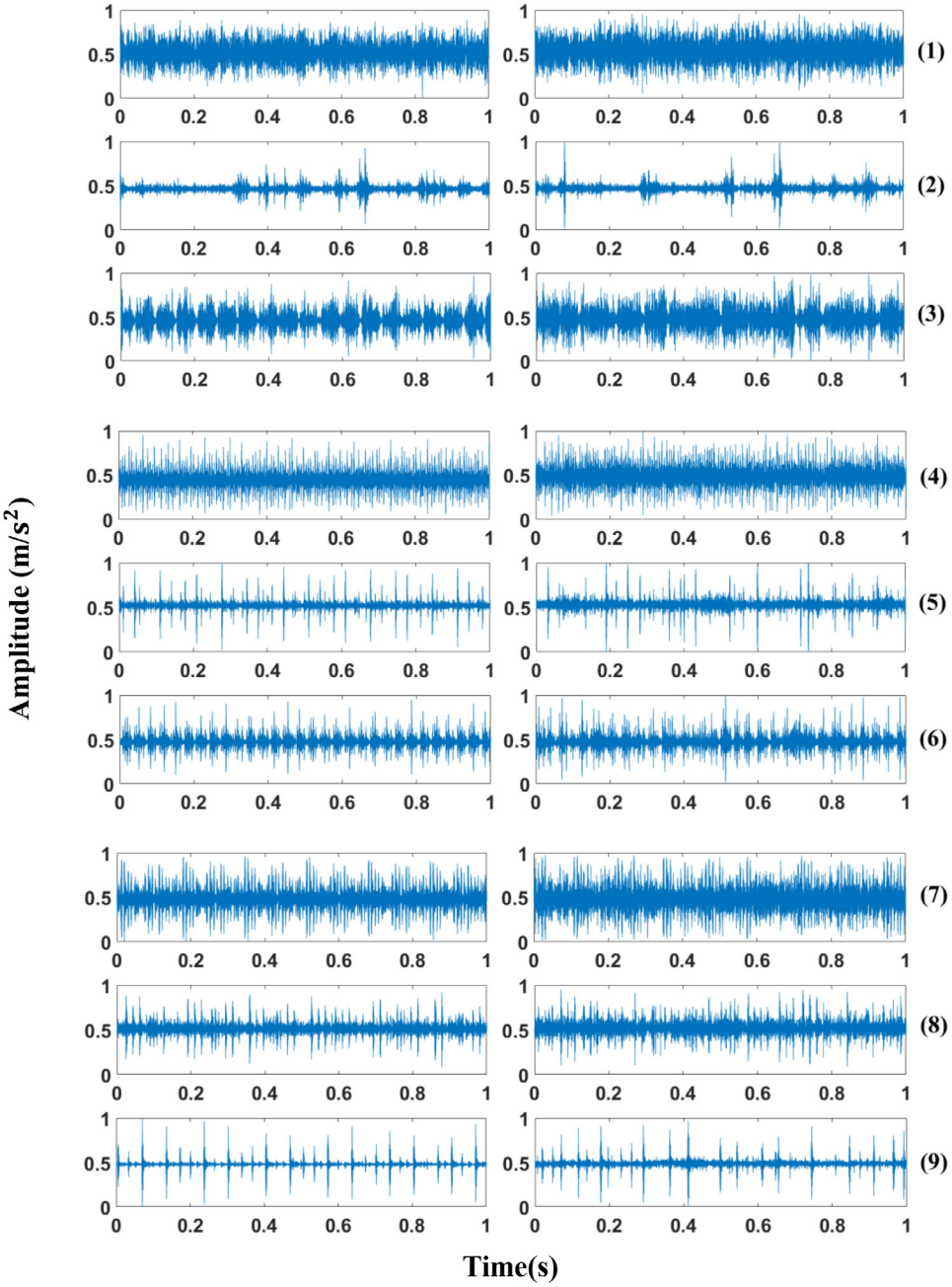


**Figure 9.** The time spectrum (1 s) comparison of the raw samples (left) and the generated samples with WGGAN-DAE (right) in nine rolling bearing faulty conditions.

In table 6, the accuracy rate is our most common evaluation index, and it is easy to understand. It is the ratio of correctly classified samples to total samples. In general, the higher the accuracy rate, the better the classifier.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{26}$$

The accuracy rate is calculated in the equation (26), where TP is the true positive, TN is the true negative, FP is the false positive, and FN is the false negative. Figure 12 gives the classification confusion matrix of the proposed method for the first trial.

**Table 3.** The RMSE of raw and generated data in time domain.

| Label | WGAN | The proposed method |
|---|---|---|
| 1 | 0.1388 | 0.0742 |
| 2 | 0.1293 | 0.0541 |
| 3 | 0.1594 | 0.0952 |
| 4 | 0.2004 | 0.1480 |
| 5 | 0.1344 | 0.0681 |
| 6 | 0.1166 | 0.0776 |
| 7 | 0.1720 | 0.1218 |
| 8 | 0.1600 | 0.1081 |
| 9 | 0.1634 | 0.0619 |

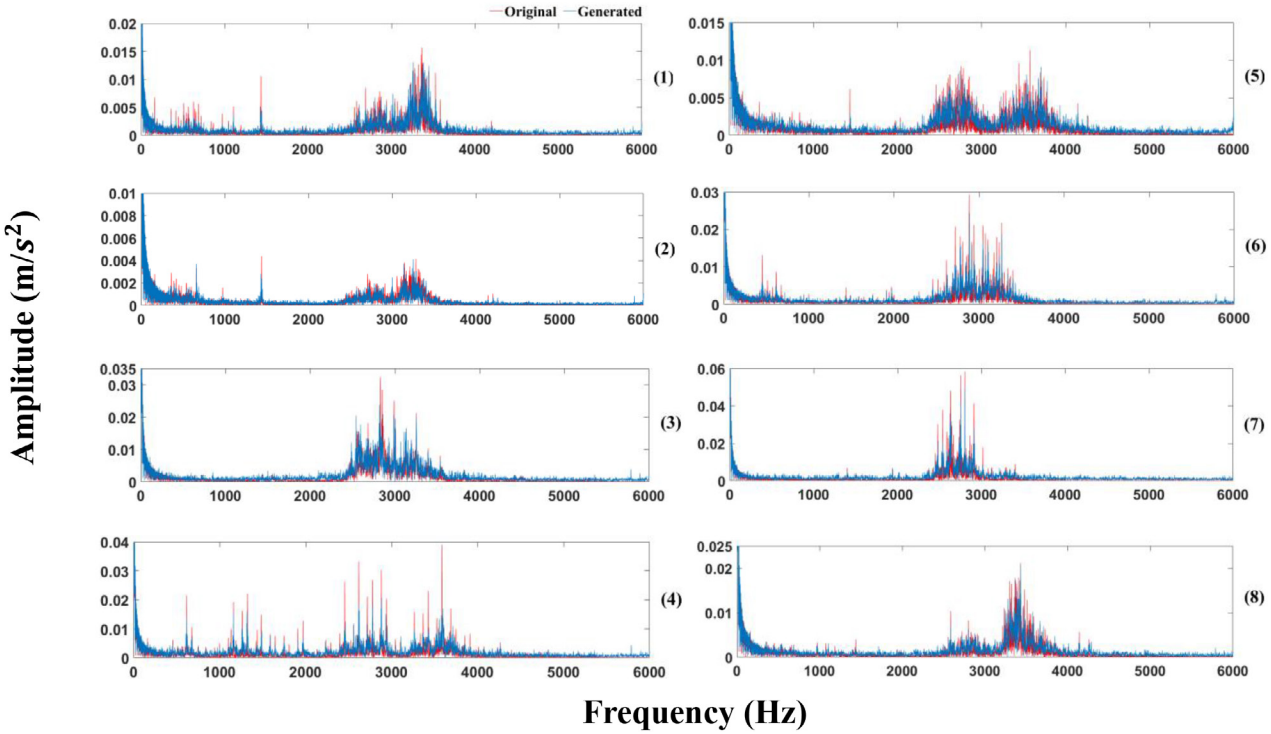


**Figure 10.** The frequency spectrum diagram comparison of the raw samples (red) and the generated samples with WGGAN-DAE (blue) in eight rolling bearing faulty operating conditions.

**Table 4.** The RMSE of raw and generated data in frequency domain.

| Label | WGAN ($10^{-4}$) | The proposed method ($10^{-4}$) |
|---|---|---|
| 1 | 7.1813 | 6.5482 |
| 2 | 8.9548 | 1.7042 |
| 3 | 9.5543 | 4.5493 |
| 4 | 14.0 | 7.0641 |
| 5 | 8.6908 | 3.2799 |
| 6 | 5.9950 | 4.1229 |
| 7 | 9.3376 | 7.4630 |
| 8 | 8.8358 | 4.3427 |
| 9 | 10.0 | 2.6281 |

Table 7 gives the precision and recall rates of different deep learning methods for the first trial. The precision rate represents the proportion of actual cases in an example divided into positive cases, and is calculated in equation (27). The recall rate is a measure of coverage, representing the proportion of pairs in all positive examples, and is calculated in equation (28).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{27}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{28}$$

$$F\text{-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{29}$$

The $F$-measure is the comprehensive evaluation index, called the weighted harmonic mean of the precision rate and recall rate, and is calculated in equation (29). As shown in figure 13, the $F$-measure values of different methods are compared.

In the experiment, different machine learning methods (BPNN, SVM, RF, DAE) are selected to avoid too much deviation in the classification results due to the problem of having a single classifier. The main parameters of the proposed method are available in table 8. The main parameters of the other five methods are described as follows:

- Method 1 (BPNN): the architecture is 800–1000–16. The learning rate is 0.01 and the iteration number is 700.
- Method 2 (SVM): the penalty factor is 50 and the radius of the kernel function is 0.2. Each of them is determined through a 10-fold cross-validation.
- Method 3 (RF): the number of trees grown is 100, and the number of predictors sampled for splitting at each node is 2.
- Method 4 (DAE): the architecture is 800–400–400–400. The sparsity parameter is 0.3 and the iteration number is 200.
- Method 5 (WGAN): the architecture is 800–400–400–400. The dimension of noise vector is 100. The learning rate is 0.0001 and the hyper parameter of Adam is 0.5. The iteration number is 2000.

In the experiment, the learning rate of the proposed method is selected as [0.0001, 0.0002, 0.0003, 0.0004, 0.0005, 0.0006, 0.0007, 0.0008, 0.0009]. Figure 14 shows the

**Table 5.** Sample distribution of the ten conditions.

| Conditions | Dataset A (raw data) | | Dataset B (synthetic data with WGAN) | | Dataset C (synthetic data with WGGAN-DAE) | | Label |
|---|---|---|---|---|---|---|---|
| | | Training/testing samples | | Training/testing samples | | Training/testing samples | |
| Normal | 100 | 50 | 100 | 50 | 100 | 50 | 1 |
| 0.007/ball | 10 | 5 | 100 | 50 | 100 | 50 | 2 |
| 0.014/ball | 10 | 5 | 100 | 50 | 100 | 50 | 3 |
| 0.028/ball | 10 | 5 | 100 | 50 | 100 | 50 | 4 |
| 0.007/inner race | 10 | 5 | 100 | 50 | 100 | 50 | 5 |
| 0.014/inner race | 10 | 5 | 100 | 50 | 100 | 50 | 6 |
| 0.021/inner race | 10 | 5 | 100 | 50 | 100 | 50 | 7 |
| 0.007_3/outer race | 10 | 5 | 100 | 50 | 100 | 50 | 8 |
| 0.007_12/outer race | 10 | 5 | 100 | 50 | 100 | 50 | 9 |
| 0.021_12/outer race | 10 | 5 | 100 | 50 | 100 | 50 | 10 |

**Table 6.** The diagnosis results of the methods.

| Method | Size of each sample | Diagnosis accuracy (%) |
|---|---|---|
| 1 BPNN (with dataset A) | 800 | 51.01 ± 4.475 (290/570) |
| 2 SVM (with dataset A) | 800 | 62.11 ± 2.531 (354/570) |
| 3 RF (with dataset A) | 800 | 64.21 ± 2.704 (366/570) |
| 4 DAE (with dataset A) | 800 | 68.42 ± 2.439 (389/570) |
| 5 WGAN (with dataset B) | 800 | 75.49 ± 1.807 (2264/3000) |
| 6 The proposed method (with dataset C) | 800 | 86.60 ± 1.012 (2598/3000) |

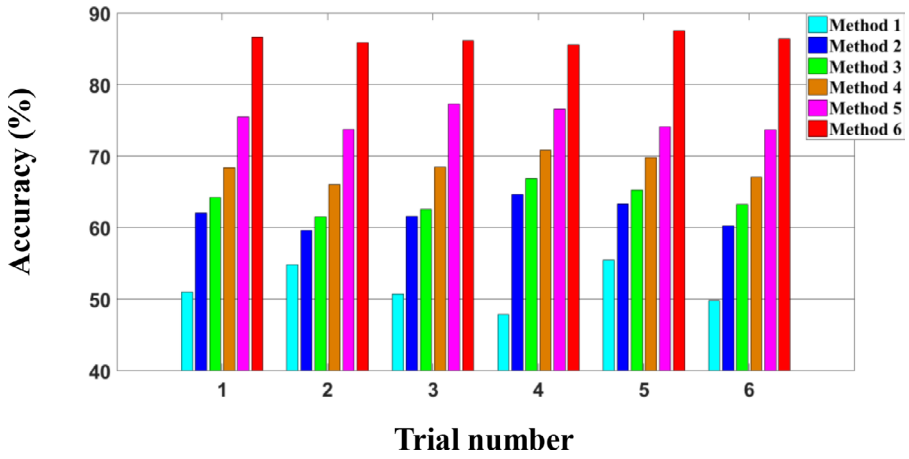


**Figure 11.** Diagnosis results of different methods for six trials.

relationship between the RMSE and the learning rate of the proposed method. It can be found that the optimal learning rate of the proposed method is 0.0002. Figure 15 shows the accuracy rate as we increase the number of hidden layers (from 1 to 3) and the number of units per hidden layer (from 100 to 800). It can be found that the optimal architecture of the proposed method is selected as 800–400–400–400. Figure 16 shows the accuracy rate as we increase parameter values of $\beta$ (candidate set [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]) and $\rho$ (candidate set [1, 2, 3, 4, 5, 6, 7, 8, 9]). It can be observed clearly that the accuracy is sensitive to the sparse parameter $\rho$. The optimal parameter $\beta$ and $\rho$ of the proposed method is selected as [0.3, 3].

Figure 17 gives the loss error curves of the discriminator in the proposed method and standard WGAN as we increase the iteration number. It can be found that the loss error of the discriminator gets closer to 1 more quickly in the proposed method. Furthermore, the discriminant probability is only close to 0.5 in WGAN at the end. So it can be found that the proposed method converges faster.

## 5. Engineering application

### *5.1. The data description of the electrical locomotive bearing experiment*

Bearings are the most widely used component in rotating machinery. In this paper, the proposed method is used to diagnose electrical locomotive bearing faults. Figure 18 gives the electrical locomotive bearing testing device. Four kinds of

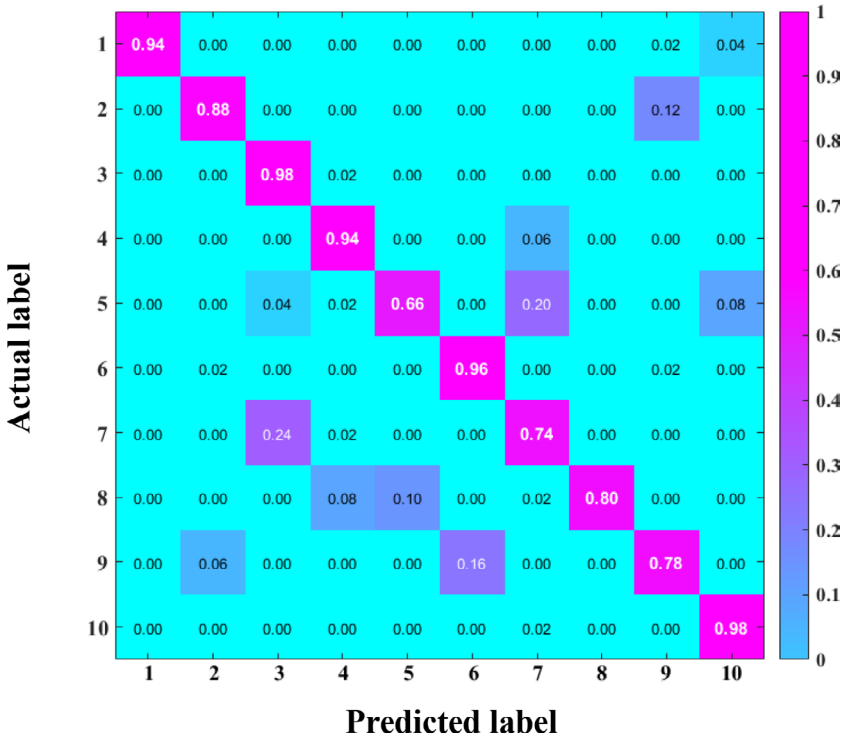


**Figure 12.** Multiclass confusion matrix of the proposed method.

**Table 7.** Precision and recall rate using different deep learning methods for the first trial.

| | SVM (%) | | RF (%) | | DAE (%) | | WGAN (%) | | The proposed method (%) | |
|---|---|---|---|---|---|---|---|---|---|---|
| Label | Precision | Recall | Precision | Recall | Precision | Recall | Precision | Recall | Precision | Recall |
| 1 | 55.56 | 100 | 14.83 | 94 | 70.15 | 94 | 100 | 98 | 100 | 94 |
| 2 | 28.57 | 80 | 100 | 20 | 100 | 20 | 67.65 | 46 | 91.67 | 88 |
| 3 | 0 | 0 | 66.67 | 80 | 50 | 20 | 70.15 | 94 | 77.78 | 98 |
| 4 | 0 | 0 | 0 | 0 | 33.34 | 20 | 94.73 | 72 | 88.68 | 94 |
| 5 | 0 | 0 | 0 | 0 | 100 | 60 | 80 | 48 | 86.84 | 66 |
| 6 | 14.29 | 20 | 75 | 60 | 17.86 | 40 | 49.32 | 72 | 85.71 | 96 |
| 7 | 0 | 0 | 47.62 | 20 | 0 | 0 | 70.41 | 84 | 71.15 | 74 |
| 8 | 30.77 | 80 | 0 | 0 | 75 | 60 | 94.12 | 64 | 100 | 80 |
| 9 | 0 | 0 | 90.9 | 20 | 33.34 | 20 | 81.72 | 72 | 82.98 | 78 |
| 10 | 0 | 0 | 97.56 | 80 | 19.87 | 60 | 64.86 | 96 | 89.09 | 98 |

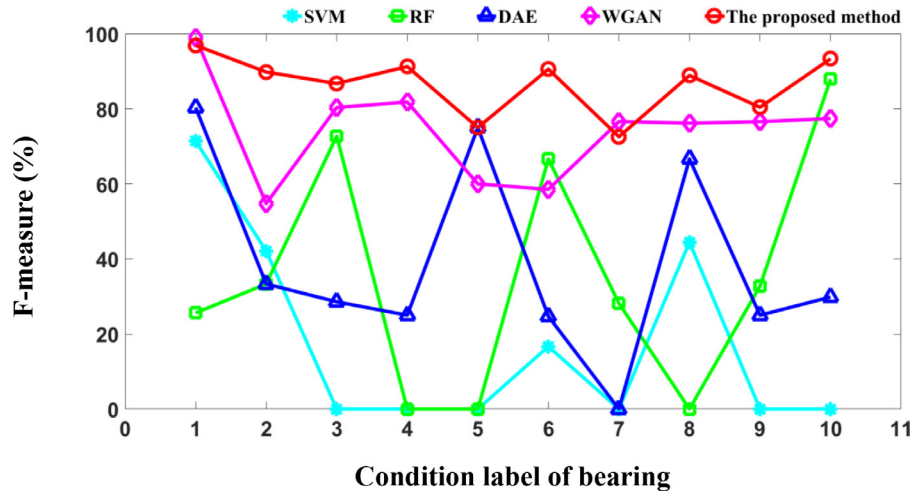


**Figure 13.** *F*-measures of the proposed method, WGAN, standard DAE, RF, and SVM.

**Table 8.** Parameters of the proposed method for rolling bearing fault diagnosis.

| Description | Symbol | Value |
|---|---|---|
| Units of the input layer | $In$ | 800 |
| Number of hidden layers | $n$ | 3 |
| Units of the first hidden layer | $h1$ | 400 |
| Units of the second hidden layer | $h2$ | 400 |
| Units of the third hidden layer | $h3$ | 400 |
| Sparsity parameter of deep auto-encoder | $\beta$ | 0.3 |
| Sparse penalty factor of deep auto-encoder | $\rho$ | 3 |
| Learning rate of WGGAN-DAE | $\alpha$ | 0.0002 |
| Update times of generator | $lg$ | 1 |
| Update times of discriminator | $ld$ | — |
| Size of training data per batch | $m$ | 64 |
| Dimension of noise vector | $z$ | 100 |
| Coefficient of gradient penalty | $\lambda$ | 0.25 |
| Hyper parameter 1 of Adam | $\gamma_1$ | 0.5 |
| Hyper parameter 2 of Adam | $\gamma_2$ | 0.1 |

faulty bearings are shown in figure 19. The vibration acceleration signal is collected at a frequency of 12.8 kHz. More parameters of the electrical locomotive bearings are listed in table 9.

In the paper, six bearing operating conditions are created, and the details are listed in table 10. Sample points are 128 000 in normal data and 12 800 in each group of fault data. There is a serious imbalance between normal and fault data. The raw vibration signals of the six bearing conditions are shown in figure 20.

### 5.2. Qualitative analysis of fault data generation samples

In this study, each class of data value is normalized to [0, 1], which means the range of each class of data point is 0 to 1, but the law of data distribution does not change, just like the original. In order to compare the raw data sample and the generated extended sample in more detail, 12 800 sample points are selected to compare the feature distributions in each of the two samples. For the convenience of data distribution comparison, the raw data are also normalized to [0, 1]. The time spectrum diagram of the samples generated by WGGAN-DAE and raw samples are shown in figure 21.

It can be seen that in figure 21, the feature distribution of the sample generated by WGGAN-DAE is very close to the raw sample, so the training generator model has a great effect. In order to more specifically show the fitting degree of the original time domain signal and the generated time domain signal, the RMSE is used in this study.

In table 11, the labels 1–6 correspond to fault data (slight outer race defect, serious outer race defect, inner race defect, roller defect, compound faults (outer and inner races), compound faults (outer race and roller)) in turn. As we can see, every RMSE of WGGAN-DAE is obviously higher than WGAN.

Next, six classes of the raw sample and the sample generated by WGGAN-DAE will be compared in the frequency spectrum diagram, at a sampling frequency of 12.8 kHz. As shown in figure 22, data points of the sample within a cycle are selected.

The above is the frequency spectrum diagram comparison of the raw sample and the generated sample of nine classes of fault data at a sampling frequency of 12.8 kHz. It can be seen that the generated sample (blue) is very close to the raw sample (red) on the frequency spectrum diagram. In order to more specifically show the fitting degree of the blue distribution and the red distribution, the RMSE is used in this paper.

In table 12, the labels 1–6 correspond to fault data (slight outer race defect, serious outer race defect, inner race defect, roller defect, compound faults (outer and inner races), compound faults (outer race and roller)) in turn. As we can see, every RMSE of WGGAN-DAE is obviously higher than WGAN.

### 5.3. The diagnosis results and analysis of the experiment

As shown in table 13, dataset A is raw data. In A, the normal condition consists of 160 samples, and each sample contains 800 data points. The random 100 raw samples are used for training and the 60 raw samples for testing. Each fault condition consists of 16 samples, and each sample contains 800 data points. The random ten raw samples are used for training and the six raw samples for testing. There is a serious imbalance between normal and fault data in dataset A. Dataset B is synthetic data with WGAN, and dataset C is synthetic data with WGGAN-DAE. In B and C, each condition consists of 160 samples, and each sample contains 800 data points. The random 100 generated samples of each condition are used for training and the 60 raw samples for testing. In order to better analyze the results, the ten different conditions are labelled 1 to 7.

BPNN, SVM, RF, and DAE are used respectively in dataset A. WGAN is used in dataset B. WGGAN-DAE is used in dataset C. The classifiers are trained with training samples at first, and are finally tested with testing samples. In order to compare the accuracy of a little raw data and a large amount of generated data more cleanly, the data was not processed too much before the experiment, to minimize other unnecessary impacts on the results.

In order to show the stability of the proposed method, each method is tried six times. The testing accuracy rates of the six methods are listed in table 14. As shown in figure 23, the testing accuracy of the proposed method in each trial is 90.47% (380/420), 88.57% (372/420), 91.90% (386/420), 89.52% (376/420), 91.43% (384/420), 89.76% (377/420). From table 14, it is observed that the average accuracy of the proposed method is 90.48% (2280/2520), higher than BPNN, SVM, RF, and DAE using the raw data, which are 90.48% (335/576), 66.67% (384/576), 71.88% (414/576), 89.76% (430/576) respectively. In the same amount of generated data,

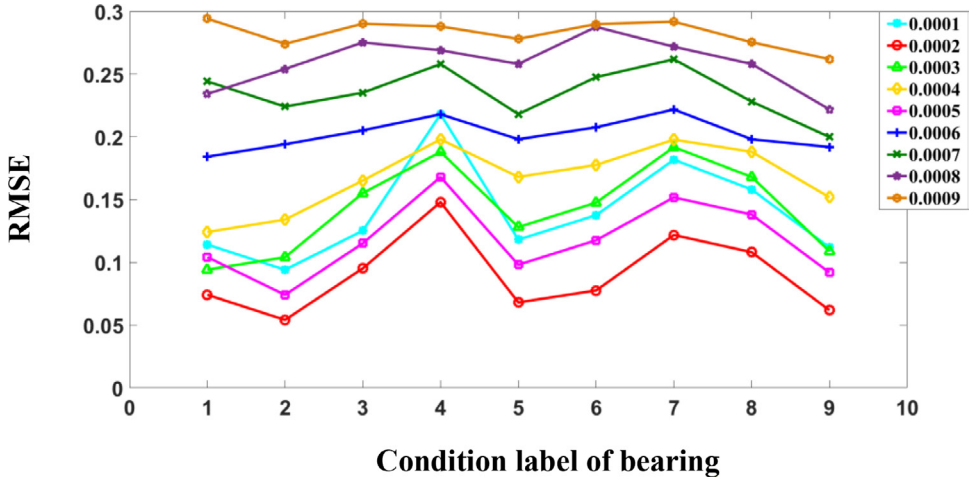


**Figure 14.** The relationship between the RMSE and the learning rate of the proposed method.

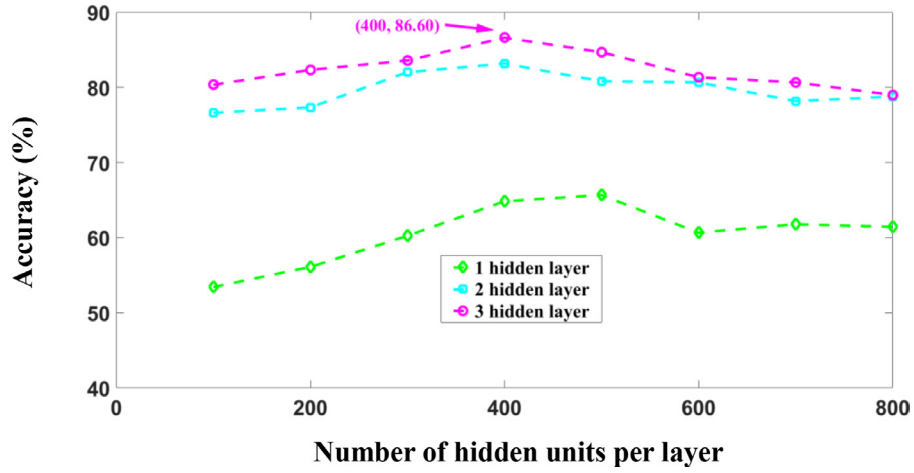


**Figure 15.** The relationship between accuracy and the proposed deep architecture.

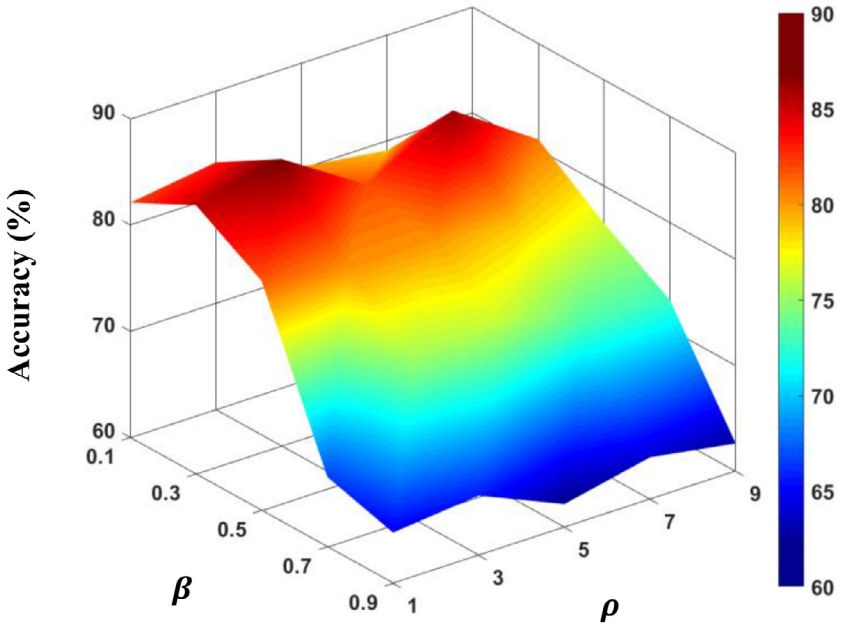


**Figure 16.** The relationship between accuracy and the parameter set ($\beta$, $\rho$).

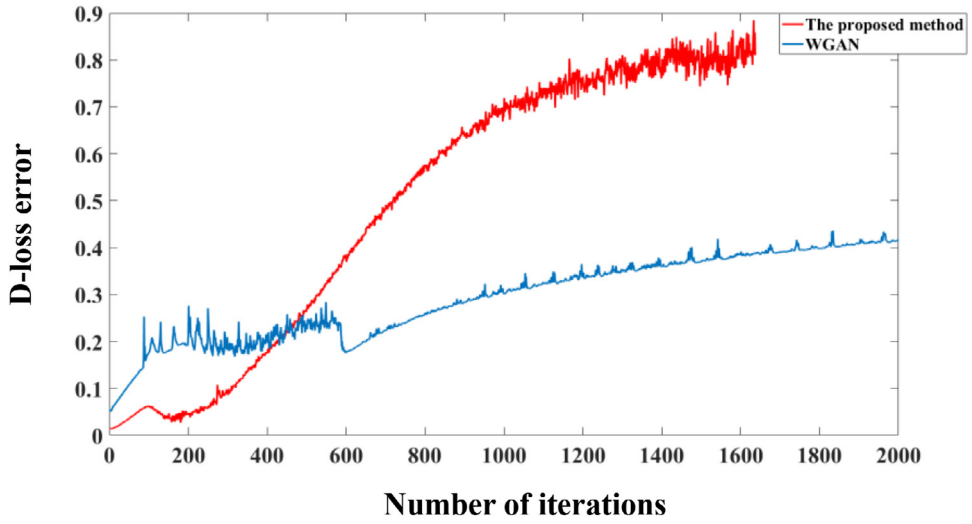


**Figure 17.** The loss function error curves of the proposed method and WGAN.

**Figure 18.** Electrical locomotive bearing testing device.

**Figure 19.** Faults in the electrical locomotive bearings are: slight inner race defect, roller defect, slight outer race defect, and serious outer race defect, in turn.

**Table 9.** Parameters of the electrical locomotive bearings.

| Parameter | Value |
|---|---|
| Bearing specs | 552732QT |
| Outer race diameter | 290 mm |
| Roller diameter | 34 mm |
| Roller number | 17 |
| Inner race diameter | 160 mm |

**Table 10.** Description of the electrical locomotive bearing operation conditions.

| Conditions | Motor speed (rpm) | Sample points |
|---|---|---|
| Normal | 490 | 128 000 |
| Slight outer race defect | 490 | 12 800 |
| Serious outer race defect | 481 | 12 800 |
| Inner race defect | 498 | 12 800 |
| Roller defect | 531 | 12 800 |
| Compound faults (outer and inner races) | 525 | 12 800 |
| Compound faults (outer race and roller) | 521 | 12 800 |

the diagnosis result of the proposed method is much higher than, WGAN which is 84.09% (2119/2520). So the quality of data generated by WGAN is not as good as that generated by the proposed method.

In table 14, the accuracy is our most common evaluation index, and it is easy to understand. Generally, the higher the accuracy, the better the classifier. Figure 24 gives the classification confusion matrix of the proposed method for the first trial. The precision and recall rates of different deep learning methods for the first trial are listed in table 15.

The $F$-measure is the weighted harmonic mean of precision and recall. As shown in figure 25, the $F$-measure values of different methods are calculated.

In this study, different machine learning methods (BPNN, SVM, RF, DAE) are selected to avoid too much deviation in the classification results due to the problem of having a

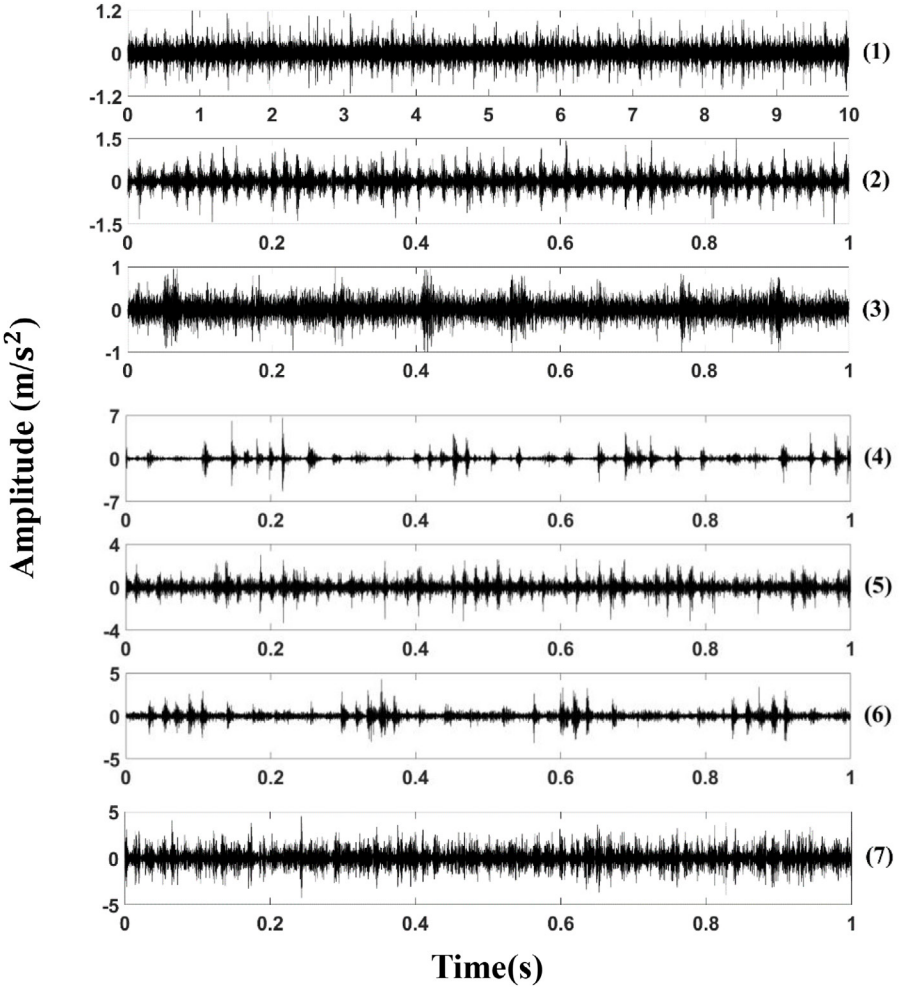


**Figure 20.** Vibration signals of the seven electrical locomotive bearing conditions: (1) normal condition; (2) slight outer race defect; (3) serious outer race defect; (4) inner race defect; (5) roller defect; (6) compound faults (outer and inner races); (7) compound faults (outer race and roller).

single classifier. The main parameters of the proposed method are listed in table 16. The main parameters of the other five methods are described as follows:

- Method 1 (BPNN): the architecture is 800–1000–16. The learning rate is 0.01 and the iteration number is 700.
- Method 2 (SVM): the penalty factor is 50 and the radius of the kernel function is 0.2. Each of them is determined through a 10-fold cross-validation.
- Method 3 (RF): the number of trees grown is 500, and the number of predictors sampled for splitting at each node is 2.
- Method 4 (DAE): the architecture is 800–400–400–400. The sparsity parameter is 0.1 and the iteration number is 400.
- Method 5 (WGAN): the architecture is 800–400–400–400. The dimension of noise vector is 100. The learning rate is 0.0003 and the hyper parameter of Adam is 0.5. The iteration number is 2500.

In the experiment, the learning rate of the proposed method is selected as [0.0001, 0.0002, 0.0003, 0.0004, 0.0005, 0.0006, 0.0007, 0.0008, 0.0009]. Figure 26 shows the relationship between the RMSE and the learning rate of the proposed method. It can be found that the optimal learning rate of the proposed method is 0.0005. Figure 27 shows the accuracy rate as we increase the number of hidden layers (from 1 to 3) and the number of units per hidden layer (from 100 to 800). It can be found that the optimal architecture of the proposed method is selected as 800–400–400–400. Figure 28 shows the accuracy rate as we increase parameter values of $\beta$ (candidate set [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]) and $\rho$ (candidate set [1, 2, 3, 4, 5, 6, 7, 8, 9]). It can be observed clearly that the accuracy is sensitive to the sparse parameter $\rho$. The optimal parameter $\beta$ and $\rho$ of the proposed method is selected as [0.1, 1].

In this paper, two datasets are used to verify the experiment. The vibration data of rolling bearings come from the Electrical Engineering Lab at Case Western Reserve University, and the

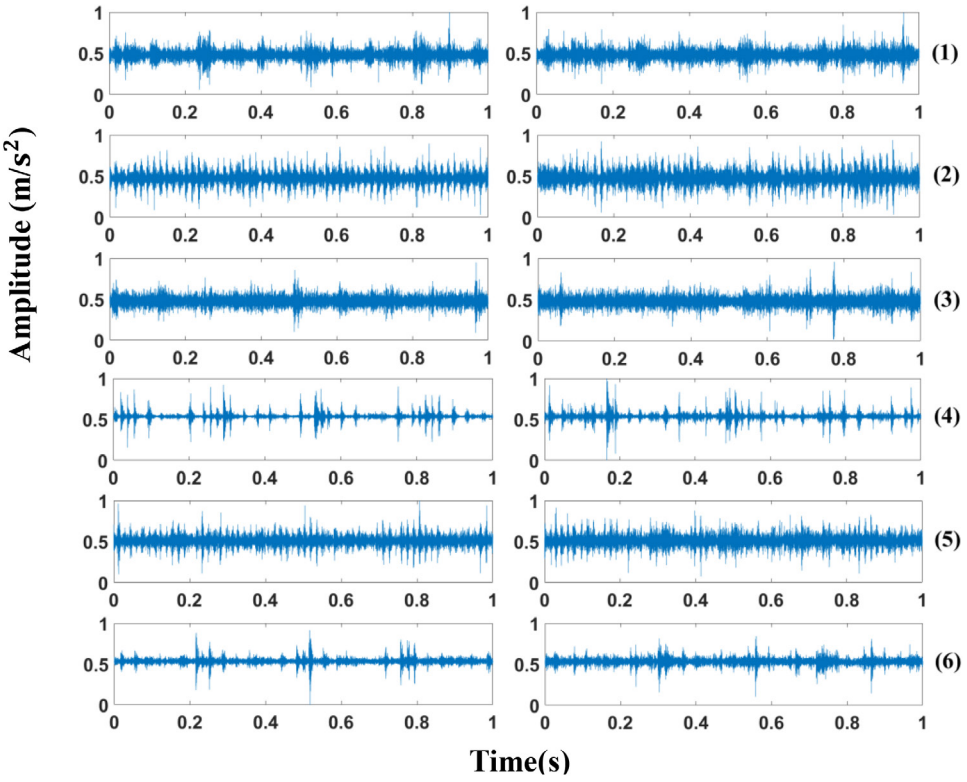


**Figure 21.** The time spectrum (1 s) comparison of the raw samples (left) and the generated samples with WGGAN-DAE (right) in six electrical locomotive bearing faulty conditions.

**Table 11.** The RMSE of raw and generated data in time domain.

| Label | WGAN | The proposed method |
|---|---|---|
| 1 | 0.1543 | 0.1117 |
| 2 | 0.2413 | 0.1201 |
| 3 | 0.0933 | 0.0867 |
| 4 | 0.0633 | 0.0576 |
| 5 | 0.1050 | 0.0947 |
| 6 | 0.0756 | 0.0582 |

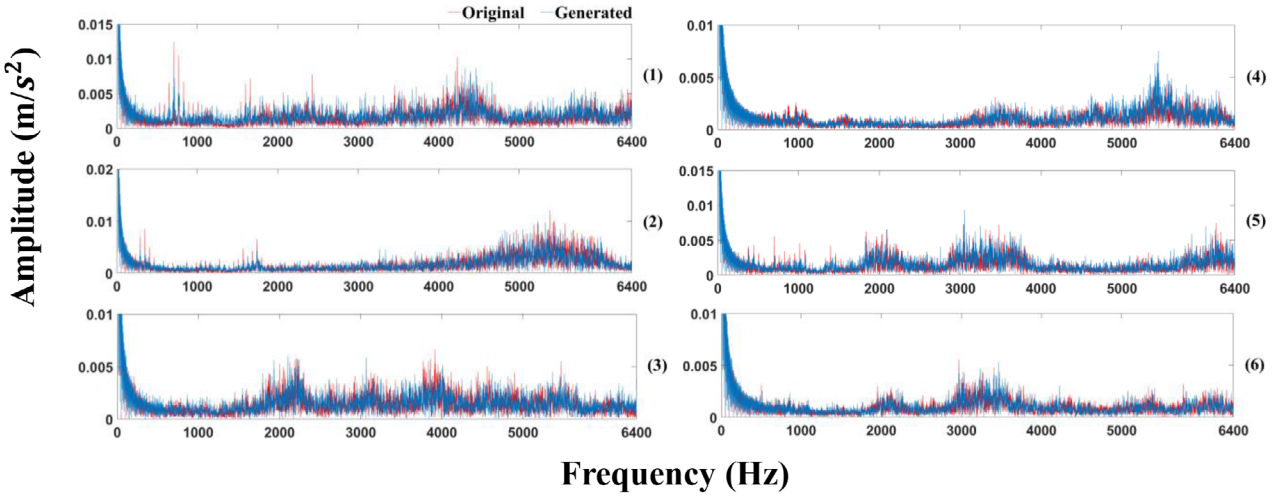


**Figure 22.** The frequency spectrum diagram comparison of the raw samples (red) and the generated samples with WGGAN-DAE (blue) in six electrical locomotive bearing faulty conditions.

**Table 12.** The RMSE of raw and generated data in frequency domain.

| Label | WGAN ($10^{-4}$) | The proposed method ($10^{-4}$) |
|---|---|---|
| 1 | 6.1796 | 3.5617 |
| 2 | 7.0027 | 3.5673 |
| 3 | 5.3502 | 2.5838 |
| 4 | 3.2793 | 1.7051 |
| 5 | 7.1260 | 2.8309 |
| 6 | 4.0119 | 1.8014 |

**Table 13.** Sample distribution of the seven conditions.

| | Dataset A (raw data) | | Dataset B (synthetic data with WGAN) | | Dataset C (synthetic data with WGGAN-DAE) | | |
|---|---|---|---|---|---|---|---|
| Conditions | Training/testing samples | | Training/testing samples | | Training/testing samples | | Label |
| Normal condition | 100 | 60 | 100 | 60 | 100 | 60 | 1 |
| Slight outer race defect | 10 | 6 | 100 | 60 | 100 | 60 | 2 |
| Serious outer race defect | 10 | 6 | 100 | 60 | 100 | 60 | 3 |
| Inner race defect | 10 | 6 | 100 | 60 | 100 | 60 | 4 |
| Roller defect | 10 | 6 | 100 | 60 | 100 | 60 | 5 |
| Compound faults (outer and inner races) | 10 | 6 | 100 | 60 | 100 | 60 | 6 |
| Compound faults (outer race and roller) | 10 | 6 | 100 | 60 | 100 | 60 | 7 |

**Table 14.** The diagnosis results of the methods.

| Method | Size of each sample | Diagnosis accuracy (%) |
|---|---|---|
| 1 BPNN (with dataset A) | 800 | 58.29 $\pm$ 4.77 (335/576) |
| 2 SVM (with dataset A) | 800 | 66.67 $\pm$ 3.94 (384/576) |
| 3 RF (with dataset A) | 800 | 72.00 $\pm$ 3.75 (414/576) |
| 4 DAE (with dataset A) | 800 | 74.66 $\pm$ 3.31 (430/576) |
| 5 WGAN (with dataset B) | 800 | 84.12 $\pm$ 2.47 (2119/2520) |
| 6 The proposed method (with dataset C) | 800 | 90.47 $\pm$ 1.63 (2280/2520) |

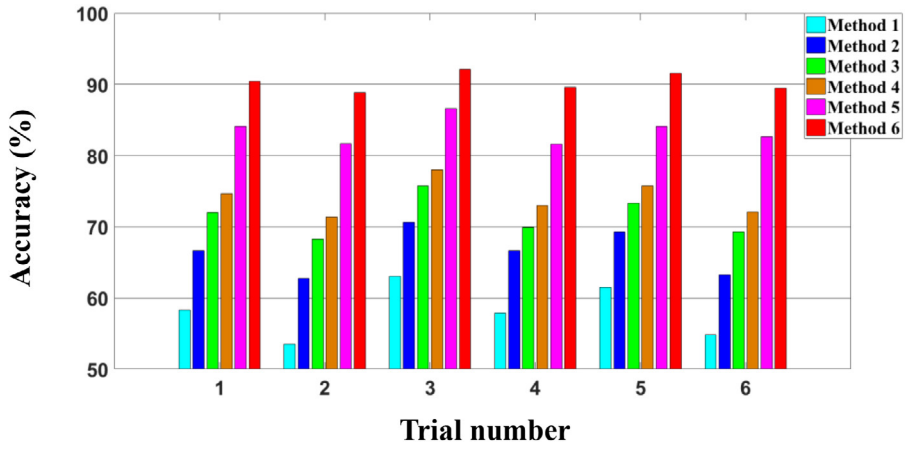


**Figure 23.** Diagnosis results of the different methods for six trials.

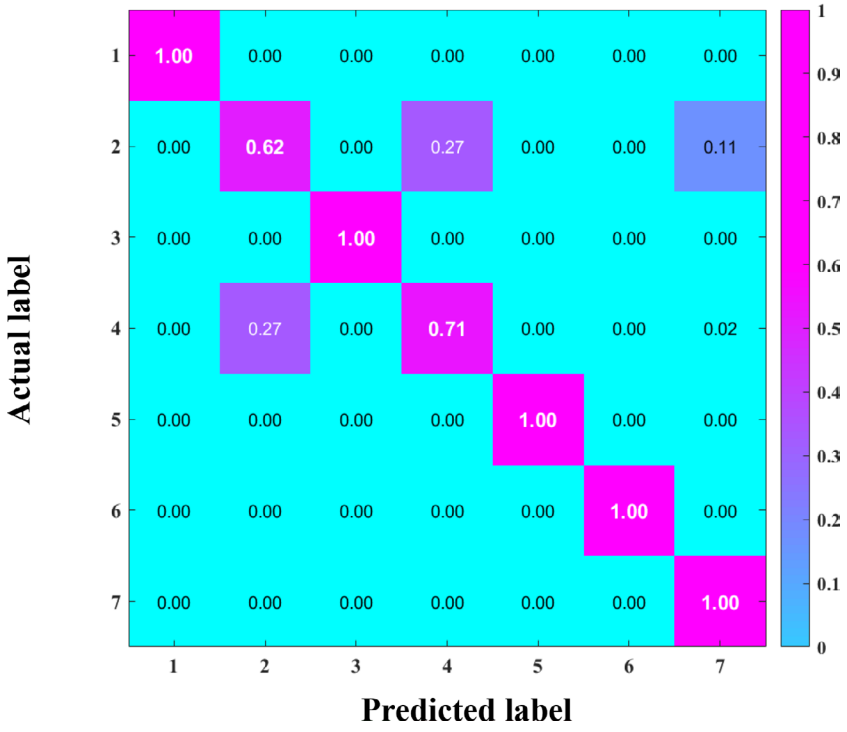


**Figure 24.** Multiclass confusion matrix of the proposed method.

**Table 15.** Precision and recall rate using different deep learning methods for the first trial.

| | SVM (%) | | RF (%) | | DAE (%) | | WGAN (%) | | The proposed method (%) | |
|---|---|---|---|---|---|---|---|---|---|---|
| Label | Precision | Recall | Precision | Recall | Precision | Recall | Precision | Recall | Precision | Recall |
| 1 | 16.67 | 100 | 20 | 100 | 27.78 | 100 | 78.13 | 100 | 100 | 100 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 62.98 | 53 | 69.66 | 62 |
| 3 | 0 | 0 | 0 | 0 | 37.50 | 60 | 77.52 | 100 | 100 | 100 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 42 | 72.45 | 71 |
| 5 | 0 | 0 | 83.33 | 100 | 75 | 60 | 100 | 96 | 100 | 100 |
| 6 | 100 | 100 | 0 | 0 | 100 | 100 | 100 | 100 | 100 | 100 |
| 7 | 0 | 0 | 100 | 80 | 0 | 0 | 66.67 | 98 | 88.50 | 100 |

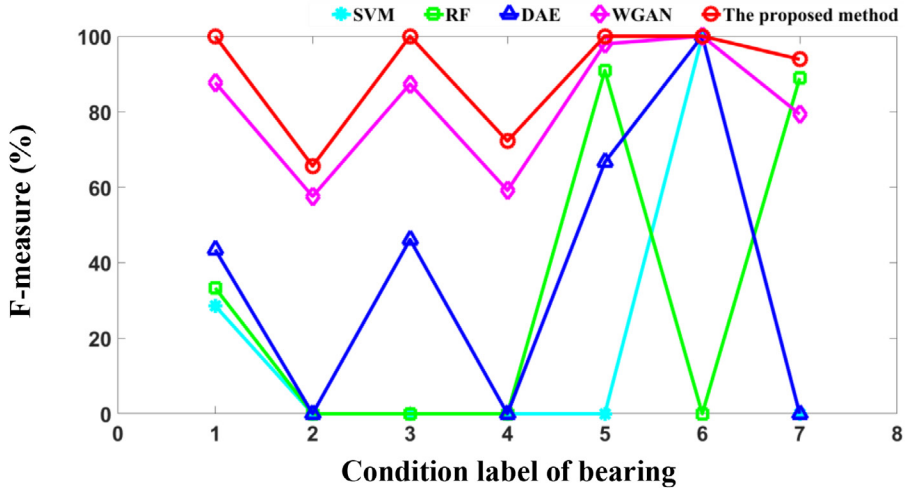


**Figure 25.** *F*-measures of the proposed method, WGAN, standard DAE, RF, and SVM.

**Table 16.** Parameters of the proposed method for rolling bearing fault diagnosis.

| Description | Symbol | Value |
|---|---|---|
| Units of the input layer | $In$ | 800 |
| Number of hidden layers | $n$ | 3 |
| Units of the first hidden layer | $h1$ | 400 |
| Units of the second hidden layer | $h2$ | 400 |
| Units of the third hidden layer | $h3$ | 400 |
| Sparsity parameter of deep auto-encoder | $\beta$ | 0.1 |
| Sparse penalty factor of deep auto-encoder | $\rho$ | 1 |
| Learning rate of WGGAN-DAE | $\alpha$ | 0.0005 |
| Update times of generator | $lg$ | 1 |
| Update times of discriminator | $ld$ | — |
| Size of training data per batch | $m$ | 64 |
| Dimension of noise vector | $z$ | 100 |
| Coefficient of gradient penalty | $\lambda$ | 0.25 |
| Hyper parameter 1 of Adam | $\gamma_1$ | 0.5 |
| Hyper parameter 2 of Adam | $\gamma_2$ | 0.1 |

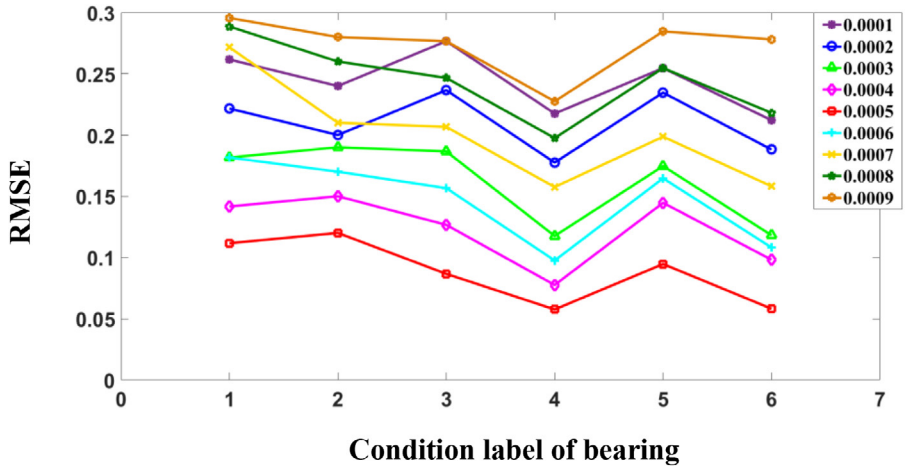


**Figure 26.** The relationship between the RMSE and the learning rate of the proposed method.

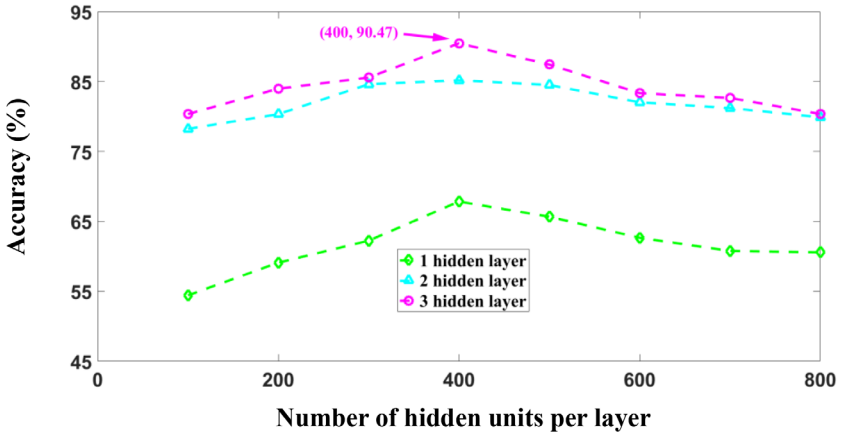


**Figure 27.** The relationship between accuracy and the proposed deep architecture.

vibration data of electrical locomotive bearing. If only one set of datasets is added to the experiment, this cannot confirm the effectiveness of the proposed method sufficiently. Two sets of different datasets can cross-validate the proposed method sufficiently.

### *5.4. The evaluation of methods through different signal-to-noise ratios*

Firstly, the dataset of the electrical locomotive bearing faults is selected to evaluate the method in different signal-to-noise ratios (SNR). In the datasets, three electrical locomotive

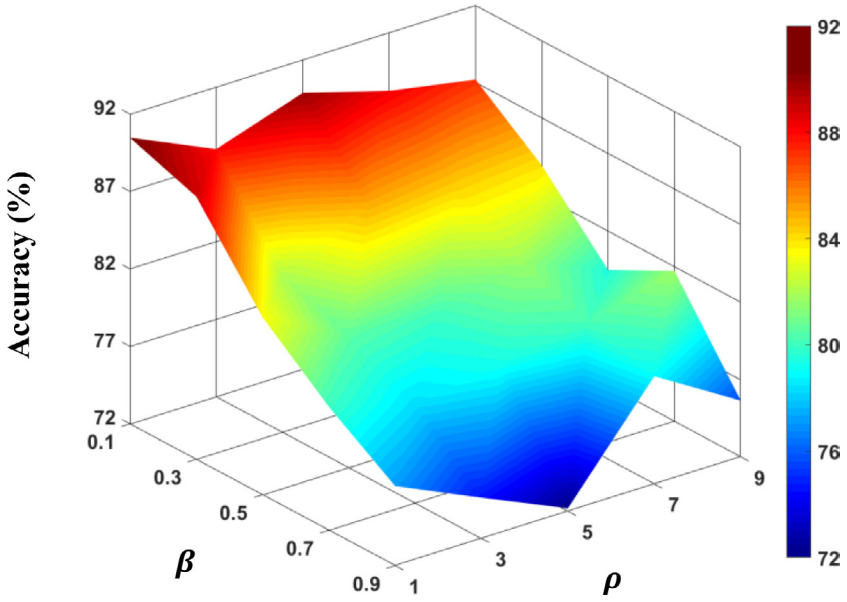


**Figure 28.** The relationship between accuracy and parameter set ($\beta$, $\rho$).

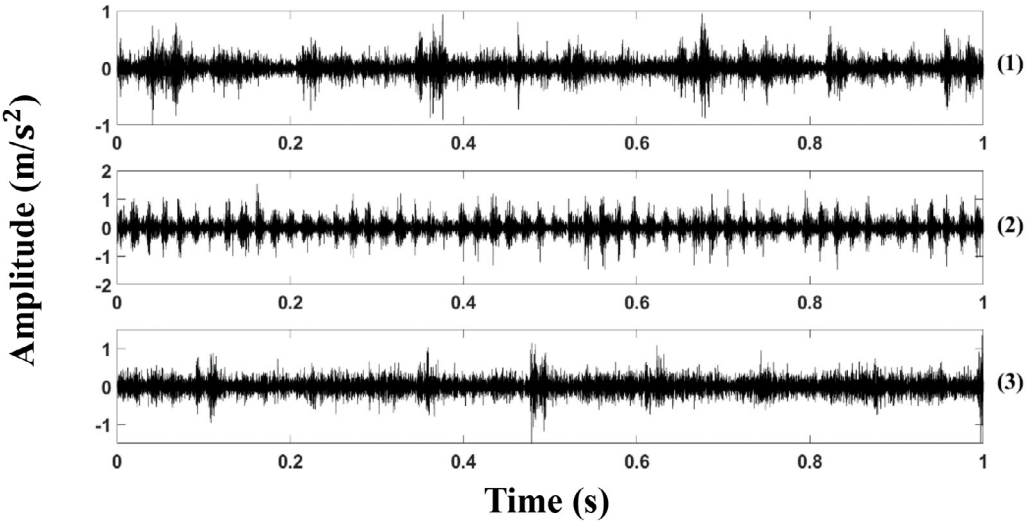


**Figure 29.** Time domain figures for electrical locomotive bearing signals: (1) normal condition; (2) slight outer race defect; (3) inner race defect.

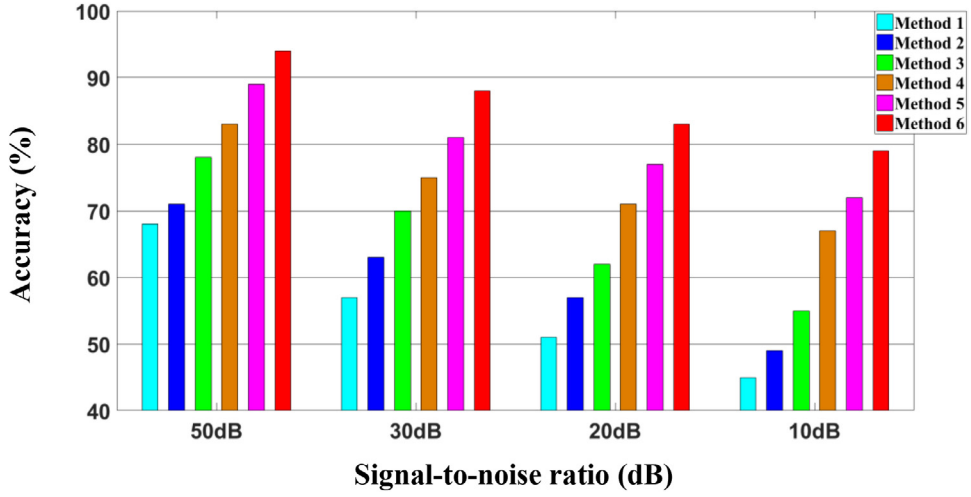


**Figure 30.** Diagnosis results of the six methods with four different SNRs.

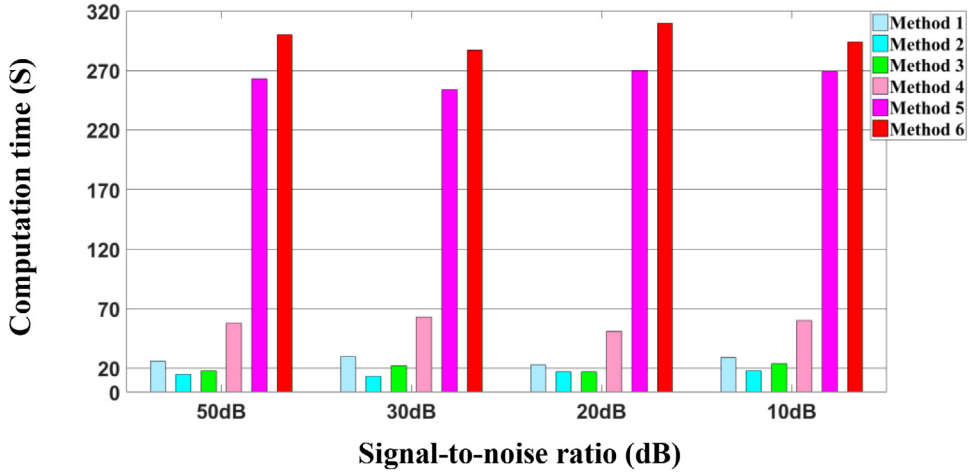


**Figure 31.** The computation times of the six methods with four different SNRs.

**Table 17.** Sample distribution of three conditions in 50 dB SNR.

| | Dataset A (raw data) | | Dataset B (synthetic data with WGAN) | | Dataset C (synthetic data with WGGAN-DAE) | | |
|---|---|---|---|---|---|---|---|
| Conditions | Training/testing samples | | Training/testing samples | | Training/testing samples | | Label |
| Normal condition | 200 | 100 | 200 | 100 | 200 | 100 | 1 |
| Slight outer race defect | 50 | 25 | 200 | 100 | 200 | 100 | 2 |
| Inner race defect | 50 | 25 | 200 | 100 | 200 | 100 | 3 |

bearing conditions (normal condition, slight outer race defect, inner race defect) are selected, as in figure 29.

SNR is defined as the ratio of signal energy to noise energy:

$$\mathrm{SNR} = \frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}} \tag{30}$$

where $P$ is an average energy. The energy of signal and noise must be measured in the same system.

The measurement unit of SNR is dB, and its calculation method is:

$$\begin{aligned} \mathrm{SNR_{db}} &= 10\log_{10}(\mathrm{SNR}) \\ &= 10\log_{10}\left(\frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}}\right) \end{aligned} \tag{31}$$

$$y = awgn\left(x, \mathrm{SNR_{db}}\right). \tag{32}$$

In equation (31), $y$ is the signal with Gaussian white noise, $x$ is the original signal and $\mathrm{SNR_{db}}$ is the customized SNR.

In this evaluation experiment, the SNR is selected as 50 dB, 30 dB, 20 dB, and 10 dB respectively. As shown in (tables 17–20), dataset A is raw data. In A, the normal condition consists of 300 samples, and each sample contains 400 data points. The random 200 raw samples are used for training and the 100 raw samples for testing. The slight outer race defect condition consists of 75 samples, and each sample contains 400 data points. The random 50 raw samples are used for training and the 25 raw samples for testing. It is the same in the inner race defect condition. Dataset B is synthetic data with WGAN, and dataset C is synthetic data with WGGAN-DAE. In B and C, each condition consists of 300 samples, and each sample contains 400 data points. The random 200 generated samples are used for training and the 100 raw samples for testing. In order to better analyze the results, the three different conditions are labelled 1 to 3.

In order to show the stability of the proposed method, each method is tried six times with a different SNR. The average testing accuracy rates of the six methods are listed in table 21. In 50 dB SNR, the accuracy based on the proposed method is 94.00%, and the accuracy based on BPNN, SVM, RF, DAE, and WGAN is 68.00%, 70.67%, 78.00%, 82.67%, 89.00%, respectively. In 30 dB SNR, the testing accuracy of the proposed method is 88.00%, higher than the other methods, which are 56.67%, 62.67%, 70.00%, 74.67%, and 81.00%, respectively. In 20 dB SNR, the testing accuracy of the proposed method is 83.00%, higher than the other methods, which are 50.67%, 56.67%, 62.00%, 70.67%, and 77.00% respectively. In 10 dB SNR, the testing accuracy of the proposed method is 79.00%, higher than the other methods, which are 44.67%, 48.67%, 54.67%, 66.67%, 72.00%, respectively.

As shown in figure 30, the testing accuracy based on the proposed method is much higher than that of traditional methods, especially with 20 dB and 10 dB. Furthermore, with the same amount of generated data, the diagnosis result of the proposed method is higher than WGAN. So the data with different proportions of noise generated by WGAN is not as good as that generated by the proposed method.

In addition, the computational time of every method is shown in figure 31. In this study, the average computation time using the proposed method is 300 s. It is much higher

**Table 18.** Sample distribution of three conditions in 30 dB SNR.

| | Dataset A (raw data) | | Dataset B (synthetic data with WGAN) | | Dataset C (synthetic data with WGGAN-DAE) | | |
|---|---|---|---|---|---|---|---|
| Conditions | Training/testing samples | | Training/testing samples | | Training/testing samples | | Label |
| Normal condition | 200 | 100 | 200 | 100 | 200 | 100 | 1 |
| Slight outer race defect | 50 | 25 | 200 | 100 | 200 | 100 | 2 |
| Inner race defect | 50 | 25 | 200 | 100 | 200 | 100 | 3 |

**Table 19.** Sample distribution of three conditions in 20 dB SNR.

| | Dataset A (raw data) | | Dataset B (synthetic data with WGAN) | | Dataset C (synthetic data with WGGAN-DAE) | | |
|---|---|---|---|---|---|---|---|
| Conditions | Training/testing samples | | Training/testing samples | | Training/testing samples | | Label |
| Normal condition | 200 | 100 | 200 | 100 | 200 | 100 | 1 |
| Slight outer race defect | 50 | 25 | 200 | 100 | 200 | 100 | 2 |
| Inner race defect | 50 | 25 | 200 | 100 | 200 | 100 | 3 |

**Table 20.** Sample distribution of three conditions in 10 dB SNR.

| | Dataset A (raw data) | | Dataset B (synthetic data with WGAN) | | Dataset C (synthetic data with WGGAN-DAE) | | |
|---|---|---|---|---|---|---|---|
| Conditions | Training/testing samples | | Training/testing samples | | Training/testing samples | | Label |
| Normal condition | 200 | 100 | 200 | 100 | 200 | 100 | 1 |
| Slight outer race defect | 50 | 25 | 200 | 100 | 200 | 100 | 2 |
| Inner race defect | 50 | 25 | 200 | 100 | 200 | 100 | 3 |

**Table 21.** The diagnosis results of the methods with different SNRs.

| | Diagnosis accuracy (%) | | | |
|---|---|---|---|---|
| Methods | 50 dB | 30 dB | 20 dB | 10 dB |
| 1 BPNN (with dataset A) | 68% (102/150) | 57% (85/150) | 51% (76/150) | 45% (67/150) |
| 2 SVM (with dataset A) | 71% (106/150) | 63% (94/150) | 57% (85/150) | 49% (73/150) |
| 3 RF (with dataset A) | 78% (117/150) | 70% (105/150) | 62% (93/150) | 55% (82/150) |
| 4 DAE (with dataset A) | 83% (124/150) | 75% (112/150) | 71% (106/150) | 67% (100/150) |
| 5 WGAN (with dataset B) | 89% (267/300) | 81% (243/300) | 77% (231/300) | 72% (216/300) |
| 6 The proposed method (with dataset C) | 94% (282/300) | 88% (264/300) | 83% (249/300) | 79% (237/300) |

than those using BPNN, SVM, RF, and DAE, which are 26 s, 15 s, 18 s, and 58 s respectively. Similarly, the computation time of WGAN is also high, at 263 s, because the proposed method and WGAN need time to generate unbalanced data first. Owing to more complex network structure of the proposed method compared to WGAN, the computation time using the proposed method is higher than that of WGAN.

## 6. Conclusions

In this paper, a novel method called the Wasserstein gradient-penalty generative adversarial network with deep auto-encoder is developed for rolling bearing intelligent fault diagnosis. This proposed method is divided into three main steps: firstly, the gradient penalty term is added to the Wasserstein generative adversarial network to enhance the stability and convergence of the network. Secondly, a deep auto-encoder network comprised of multiple auto-encoders is regarded as the discriminator. Finally, the sparse auto-encoder is placed at the end of the proposed method as the classifier to classify synthetic bearing faults.

This proposed method is applied to generate the effective rolling bearing fault data, to improve the problem of unbalanced fault data, and to analyze the rolling bearing vibration signals more accurately. The results confirm that the generated data are very close to the raw data on the time-frequency diagram, and the diagnosis results of the proposed method are better than other traditional methods and the Wasserstein generative adversarial network. Moreover, the proposed method is more stable and converges faster than the Wasserstein generative adversarial network. The author will continue to work on this subject area in future research.

## Acknowledgments

This research is supported by the National Natural Science Foundation of China (No. 51875459), the major research

plan of the National Natural Science Foundation of China (No. 91860124), and the Aeronautical Science Foundation of China (No. 20170253003).

## ORCID iDs

Jiang Hongkai https://orcid.org/0000-0001-6180-4641
Xingqiu Li https://orcid.org/0000-0002-7568-0077

## References

[1] Jiang H K, Li C L and Li H X 2013 An improved EEMD with multiwavelet packet for rotating machinery multi-fault diagnosis *Mech. Syst. Signal Process.* **36** 225–39
[2] Qian Y N, Yan R Q and Gao R X 2017 A multi-time scale approach to remaining useful life prediction in rolling bearing *Mech. Syst. Signal Process.* **83** 549–67
[3] Shan Y H, Zhou J Z and Jiang W 2019 A fault diagnosis method for rotating machinery based on improved variational mode decomposition and a hybrid artificial sheep algorithm *Meas. Sci. Technol.* **30** 055002
[4] Shao H D, Jiang H K, Li X Q and Wu S P 2018 Intelligent fault diagnosis of rolling bearing using deep wavelet auto-encoder with extreme learning machine *Knowl.-Based Syst.* **40** 1–14
[5] Yuan Z, Zhang L B and Duan L X 2018 A novel fusion diagnosis method for rotor system fault based on deep learning and multi-sourced heterogeneous monitoring data *Meas. Sci. Technol.* **29** 115005
[6] Qian W W, Li S M and Wang J R 2018 An intelligent fault diagnosis framework for raw vibration signals: adaptive overlapping convolutional neural network *Meas. Sci. Technol.* **29** 095009
[7] Ma P, Zhang H L and Fan W H 2019 A novel bearing fault diagnosis method based on 2D image representation and transfer learning-convolutional neural network *Meas. Sci. Technol.* **30** 055402
[8] Jiang H K, Li X Q and Shao H D 2018 Intelligent fault diagnosis of rolling bearings using an improved deep recurrent neural network *Meas. Sci. Technol.* **29** 065107
[9] Guo L, Li N P, Jia F, Lei Y G and Lin J 2017 A recurrent neural network based health indicator for remaining useful life prediction of bearings *Neurocomputing* **240** 98–109
[10] Shao H D, Jiang H K and Zhang X 2015 Rolling bearing fault diagnosis using an optimization deep belief network *Meas. Sci. Technol.* **26** 115002
[11] Hoang D T and Kang H J 2018 A survey on deep learning based bearing fault diagnosis *Neurocomputing* **335** 327–35
[12] Saufi S R, Ahmad Z A B and Leong M S 2018 Differential evolution optimization for resilient stacked sparse autoencoder and its applications on bearing fault diagnosis *Meas. Sci. Technol.* **29** 125002
[13] Wang J R, Li S M and Han B K 2019 Construction of a batch-normalized autoencoder network and its application in mechanical intelligent fault diagnosis *Meas. Sci. Technol.* **30** 015106
[14] Zeng N Y, Zhang H, Song B Y and Liu W B 2018 Facial expression recognition via learning deep sparse autoencoders *Neurocomputing* **273** 643–9
[15] Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial nets *Int. Conf. on Neural Information Processing Systems* (MIT)
[16] Douzas G and Bacao F 2018 Effective data generation for imbalanced learning using conditional generative adversarial networks *Exp. Syst. Appl.* **91** 464–71
[17] Wang Z R, Wang J and Wang Y R 2018 An intelligent diagnosis scheme based on generative adversarial learning deep neural networks and its application to planetary gearbox fault pattern recognition *Neurocomputing* **310** 213–22
[18] Goodfellow I J 2016 *NIPS 2016 Tutorial: Generative Adversarial Networks* (arXiv:1701.00160)
[19] Shao H D, Jiang H K, Wang F A and Zhao H W 2017 An enhancement deep feature fusion method for rotating machinery fault diagnosis *Knowl.-Based Syst.* **119** 200–20
[20] Arjovsky M, Chintala S and Bottou L 2017 Wasserstein GAN (arXiv:1701.07875)
[21] Liu Y F, Qin Z C, Wan T and Luo Z B 2018 Auto-painter: cartoon image generation from sketch by using conditional Wasserstein generative adversarial networks *Neurocomputing* **311** 78–87
[22] Zhou C S, Zhang J S and Liu J M 2018 Lp-WGAN: using Lp-norm normalization to stabilize Wasserstein generative adversarial networks *Knowl.-Based Syst.* **161** 415–24
[23] Wang S H, Xiang J W and Zhong Y T 2018 Convolutional neural network-based hidden Markov models for rolling element bearing fault identification *Knowl.-Based Syst.* **144** 65–76
[24] Jia J Y, Zha M and Lin J 2018 A multivariate encoder information based convolutional neural network for intelligent fault diagnosis of planetary gearboxes *Knowl.-Based Syst.* **160** 237–50
[25] Yan X A and Jia M P 2019 Intelligent fault diagnosis of rotating machinery using improved multiscale dispersion entropy and mRMR feature selection *Knowl.-Based Syst.* **163** 450–71
[26] Wang Y R, Jin Q and Sun G D 2019 Planetary gearbox fault feature learning using conditional variational neural networks under noise environment *Knowl.-Based Syst.* **163** 438–49
[27] Wang J R, Li S M and An Z H 2019 Batch-normalized deep neural networks for achieving fast intelligent fault diagnosis of machines *Neurocomputing* **329** 53–65
[28] Kuwada K 2010 Duality on gradient estimates and Wasserstein controls *J. Funct. Anal.* **258** 3758–74
[29] Rippl T, Munk A and Sturm A 2016 Limit laws of the empirical Wasserstein distance: Gaussian distributions *J. Multivariate Anal.* **151** 90–109
[30] Salimans T, Goodfellow I J and Zaremba W 2016 Improved techniques for training GANs (arXiv:1606.03498)