

Research Article

Research on Fault Diagnosis Model of Generative Adss Based on Improved Semisupervised Diagnosis Algorithm

Yi Qian 

College of Information Science and Technology, Taishan University, Taian 271000, Shandong, China

Correspondence should be addressed to Yi Qian; qianyi_tsu@tsu.edu.cn

Received 19 September 2021; Revised 7 November 2021; Accepted 9 November 2021; Published 2 December 2021

Academic Editor: Sadaqat ur Rehman

Copyright © 2021 Yi Qian. 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.

With the advent of the era of big data and the rapid development of deep learning and other technologies, people can use complex neural network models to mine and extract key information in massive data with the support of powerful computing power. However, it also increases the complexity of heterogeneous network and greatly increases the difficulty of network maintenance and management. In order to solve the problem of network fault diagnosis, this paper first proposes an improved semisupervised inverse network fault diagnosis algorithm; the proposed algorithm effectively guarantees the convergence of generated against network model, makes full use of a large amount of trouble-free tag data, and obtains a good accuracy of fault diagnosis. Then, the diagnosis model is further optimized and the fault classification task is completed by the convolutional neural network, the discriminant function of the network is simplified, and the generation pair network is only responsible for generating fault samples. The simulation results also show that the fault diagnosis algorithm based on network generation and convolutional neural network achieves good fault diagnosis accuracy and saves the overhead of manually labeling a large number of data samples.

1. Introduction

With the advent of the era of big data and the rapid development of deep learning and other technologies, people can mine and extract key information from massive data by using complex neural network models with the support of powerful computing power [1]. Especially in the complex heterogeneous network environment, thousands of network nodes will generate a large amount of network operation information every day [2]. However, this also increases the complexity of heterogeneous networks and increases the operating costs of network maintenance and management for operators. Some scholars have proposed a self-organizing fault diagnosis algorithm for heterogeneous networks based on AdaBoost ensemble learning, by training Bayesian, decision tree, artificial neural network, and K nearest neighbor classifiers, combined their classification results to gradually approach the real classification results, and used SMOTE technology to deal with the problem of unbalanced data in cellular networks, avoiding the low classification accuracy of

a few classes and effectively improving the diagnostic accuracy of classifiers [3]. It is undeniable that intelligent diagnosis algorithms such as RNN, LSTM, and CNN have satisfactory accuracy, but the premise of this high accuracy is to manually add labels to all samples, which will undoubtedly increase the cost of fault diagnosis [4]. Some scholars have proposed a semisupervised generation countermeasure network algorithm based on Gini regularization, which improves the convergence speed and accuracy of the model by adding Gini regularization constraints to the unsupervised loss function of the discriminator. In this algorithm, Wellside data is used as marked data, and seismic data is used as unmarked data to identify lithology, which reduces drilling cost [5]. Some scholars have also proposed a semisupervised classification algorithm based on joint training of generating countermeasure networks [6]. Firstly, two GAN models are trained once with labeled data sets. Then the unlabeled data set is input into the discriminators of the two GAN models. If the two discriminators have the same discriminating results to a

certain sample, the corresponding category label, namely, pseudo-label sample, is added to the labeled data set, the above operation is repeated until no new pseudolabel sample is generated, and the new labeled data set is used to train GAN model to complete the final classification task. Although the algorithm makes full use of unlabeled data sets, it only generates pseudolabeled samples according to whether the output results of two GAN models are the same, and its credibility directly affects the classification effect of the final GAN model.

Soft fault is the most difficult problem in analog circuits. Researchers have found a method to classify the samples randomly generated in the soft fault interval by using semisupervised support vector machine algorithm and then extract the circuit fault response data by manifold learning algorithm. Through verification, the classification method can be used for circuit soft fault diagnosis, which is quite instructive for practical application [7]. Tag data is actually difficult to collect and costs a lot. In the intelligent fault diagnosis of convolutional neural network, it often depends on labeled data. Therefore, we want to use unlabeled data. A bearing intelligent fault diagnosis method based on semisupervised convolutional neural network is proposed, and then the semisupervised convolutional neural network model is used to analyze the bearing vibration signal. Experiments show that our semisupervised convolutional neural network can train the model with unlabeled samples and improve the performance of diagnosis. Its advantages are obvious compared with common methods [8]. With the development of society and the improvement of living standards, people have higher and higher requirements for air quality in building rooms. At present, we generally use air handling devices to realize the circulation and regulation of indoor air, which is particularly important for the accurate diagnosis and troubleshooting of air conditioning equipment. Although the data-driven diagnosis method widely used in practice has its flexibility and advantages, its further implementation of advanced supervised learning algorithm is not in line with the reality because it takes too much time and effort to obtain the wrong data label [9]. Therefore, we propose a semisupervised diagnosis method based on neural network, which adopts self-training strategy. Through a large number of experiments, we believe that it can improve the generalization performance of the model and play a positive role in the development of advanced data-driven fault diagnosis system tools in the future. The artificial data generated by the generated countermeasure network has high similarity with the real data. At the same time, it can be used as a tool for data enhancement in the task of image generation. We use the developed auxiliary classifier GAN framework to generate realistic raw data. The framework can effectively avoid the gradient disappearance problem, and the category label is used to assist the training model [10]. We use a set of evaluation methods to evaluate the quality of the generated samples and confirm their performance.

The contribution of this paper is as follows. (1) In view of the current network fault diagnosis problem of hard, at first, this paper puts forward a kind of based on improved the generation of a semisupervised against network fault

diagnosis algorithms, the proposed algorithm effectively guarantees the convergence of generated against network model and makes full use of a large amount of trouble-free tag data, and obtains a good accuracy of fault diagnosis. And then we use a diagnosis model for the further optimization. (2) The generate against network based combined with convolution neural network fault diagnosis algorithm achieves good accuracy of fault diagnosis and saves the overhead of manually marking a large number of data samples.

2. Fault Diagnosis Algorithm Based on Improved Generation Countermeasure Network

2.1. System Model. Aiming at the dense heterogeneous wireless network scenario with multilevel network structure composed of high-power Hong Jizhan and low-power micro base stations [11, 12], as shown in Figure 1, this paper proposes a network fault diagnosis model based on GAN to locate faults, so as to avoid losses caused by fault propagation.

In the selection of key performance parameters used to characterize the network condition, this paper adopts the feature selection algorithm combining Reliance and mutual information, and finally selects eight key performance indexes, RSRQ_P, DCR, HO, RSRP_P, ERAB_S, SNR_UL, SNR_DL and LER, as the input parameters of the fault diagnosis model.

2.2. Network Fault Diagnosis Model Based on Improved Semisupervised Generation Countermeasure. Although the original GAN does not need classification labels in the training process, it generates false data similar to the real data, and its discriminator realizes the identification of true and false data, but it cannot classify the real data [13, 14]. Some scholars skillfully modify the discriminator of GAN and replace the output layer of discriminator network with Softmax classifier, so that it can classify real data. Assuming that the training data has K classes, the Softmax classifier has $K + 1$ classes, where the $K + 1$ class represents the false data generated by the generator, and the rest correspond to K classes in the training data, so that GAN can complete the semisupervised classification problem, and its network structure is shown in Figure 2.

The training network D maximizes the probability of sorting the labels of the training samples (maximizing $\log D(x)$), and the training network G minimizes $\log(1 - D(g(c)))$; that is, it maximizes the loss of D . In the training process, one party is fixed and the parameters of the other network are updated and iterated alternately to maximize the other party's error. Finally, G can estimate the distribution of sample data. Due to the change of discriminator structure, the loss function of semisupervised generation countermeasure network (SGAN) also changes accordingly. The loss function of generator G is shown in formula (1), where $\hat{x} = G(z)$ is the data generated by the generator, $p(\hat{x})$ is the probability distribution function of \hat{x} , and $P_d(y = K + 1|\hat{x})$ represents the probability that the generated data \hat{x} is judged as $K + 1$ class by discriminator D .

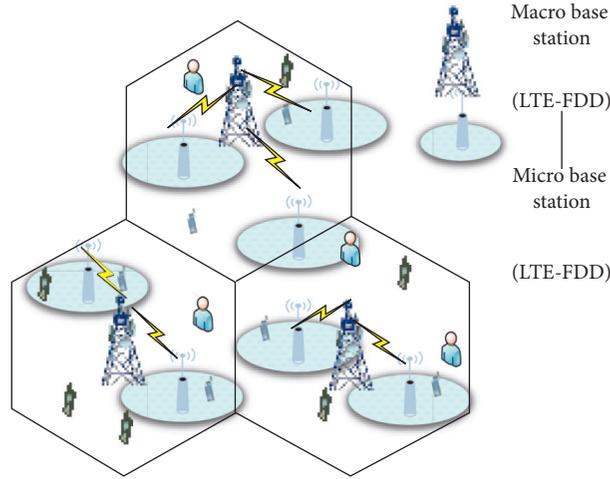


FIGURE 1: Two-tier heterogeneous wireless network model.

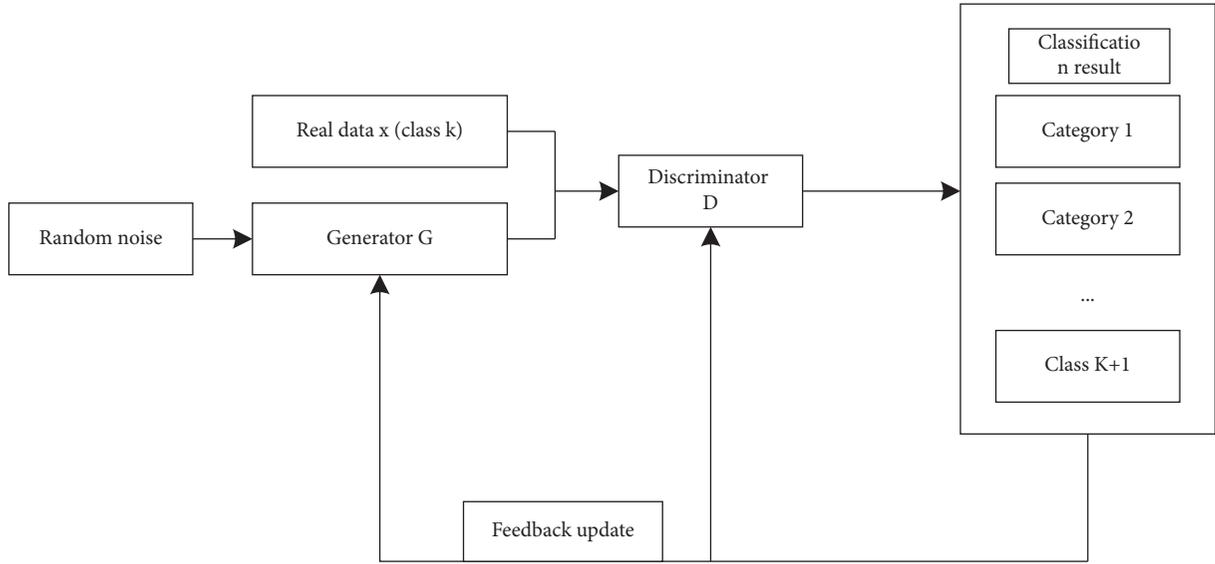


FIGURE 2: Semisupervised generation countermeasure network structure.

$$L_G = E_{\hat{x} \sim p(\hat{x})} [\log P_d(y = K + 1 | \hat{x})]. \quad (1)$$

The loss function of discriminator D can be divided into supervised loss and unsupervised loss, as shown in

$$\begin{aligned} L_D &= -E_{(x,y) \sim p(x,y)} [\log P_d(y|x)] \\ &\quad - E_{\hat{x} \sim p(\hat{x})} [\log P_d(y = K + 1 | \hat{x})] \quad (2) \\ &= L_{\text{sup}} + L_{\text{unsup}}, \end{aligned}$$

where X is the real data, Y is the category label of X , $P(X, Y)$ is the probability distribution obeyed by (X, Y) , X is the data generated by the generator, and $p(\hat{x})$ is the probability distribution obeyed by it. The loss function of the supervised part refers to the distance loss between the predicted result and the label when training the labeled part of data (x, y) , as shown in formula (3). The loss function of the unsupervised part refers to the loss when training the unlabeled part of data x and the generated data \hat{x} , that is, the loss function of the original GAN discriminator, as shown in formula (4).

$$L_{\text{sup}} = -E_{(x,y) \sim p(x,y)} [\log P_d(y|x)], y \in \{1, 2, \dots, K + 1\}, \quad (3)$$

$$L_{\text{unsup}} = -\left\{ E_{x \sim p(x)} [\log(1 - P_d(y = K + 1|x))] + E_{\hat{x} \sim p(\hat{x})} [\log P_d(y = K + 1|\hat{x})] \right\}. \quad (4)$$

Although SGAN can improve the performance of supervision tasks by learning additional unlabeled samples, the goal of gradient descent algorithm in SGAN is to find the minimum loss function point, instead of finding a Nash equilibrium that makes both the generator and the discriminator achieve the best performance. Especially when there are many input feature parameters, the difficulty of convergence of the model will greatly increase. Therefore, in this paper, two constraint functions, feature matching and compactness calculation, are added to the loss function of the generator network to stabilize the convergence of SGAN model and improve the final performance of the model. The model structure is shown in Figure 3.

Among them, the calculation formula of convolution operation is as follows:

$$y_{i^{l+1},j^{l+1}} = \sum_{i=0}^H \sum_{j=0}^w f_{i,j} \cdot x_{i^{l+1},j^{l+1}}^l \quad (5)$$

In this paper, a 3×3 convolution kernel is adopted, and in order to prevent the omission of data information, the convolution step is set to 1; that is, the distance of each convolution kernel movement is a coordinate unit, so the output meets the following requirements:

$$\begin{aligned} 0 \leq i^{l+1} \leq H^l - H + 1 = H^{l+1}, \\ 0 \leq j^{l+1} \leq W^l - W + 1 = W^{l+1}. \end{aligned} \quad (6)$$

When SGAN trains generator G , originally, it only depends on reducing the distance loss between the output result of discriminator D and the real result. That is to say, the only goal is to increase the probability that the discriminator D regards false data as true data, so it is easy to cause the generator G to be overtrained on the current discriminator D , which leads to the great influence on the generator G when the parameters of the discriminator D change, so that the convergence fluctuation of the model is large, and it is difficult to reach the Nash equilibrium point of D and G . Therefore, in this paper, an objective function $D(x, \hat{x})$ of the distance between the generated data and the real data is added to the loss function of the generator G , that is, the feature matching between the generated data and the real data, so as to make the generated data consistent with the real data as much as possible and avoid overtraining on the current discriminator. The main idea is that when the false data \hat{x} generated by the generator is input to the discriminant network and when the calculation result of each hidden layer in the discriminator is as similar as possible as the real data x is input into the discriminant network, then the L2 distance is calculated for each hidden layer result, and the distance of all layers is summed and averaged to obtain the objective function $D(x, \hat{x})$. The formula is as follows:

$$D(x, \hat{x}) = \frac{1}{l} \sum_{i=1}^l E_{x \sim p(x)} f_i(x) - E_{\hat{x} \sim p(\hat{x})} f_i(\hat{x}), \quad (7)$$

where $f_i(\cdot)$ is the output of the i -th hidden layer of the discriminator, l is the number of hidden layers, x is the real data, and \hat{x} is the generated data.

In the loss function of the generator, we add the distance function between the compactness between the real samples and the compactness between the generated samples, so as to avoid the situation that the loss function tends to be unchanged and the algorithm cannot converge correctly when the model collapses.

In this paper, the definition of compactness modeling between samples is as follows: the distance between a single sample and each sample in a small batch data set is summed and averaged to characterize the compactness between the sample and the data set. Considering the differences in dimensions of each feature in the sample, the distance between the samples adopts standardized Euclidean distance, and the distance function between the compactness between the real samples and the compactness between the generated samples is

$$\begin{aligned} \text{dis} = |D_{\text{ral}} - D_{\text{fake}}| &= \frac{1}{n} \sum_{r=1}^n f(\hat{x}) \\ &- f(x_r)_{\text{normal}} - \frac{1}{n} \sum_{f=1}^n f(\hat{x}) - f(x_f), \end{aligned} \quad (8)$$

where \hat{x} is the input data of the discriminator, X_r and X_f are the samples of the small batch real data set and the small batch generated data set respectively, n is the size of the small batch data set, and $f(\cdot)$ is the output of the hidden layer in the middle of the discriminator. The normalized Euclidean distance between two n -dimensional variables a ($x_{11}, x_{12}, \dots, x_{1n}$) and b ($x_{21}, x_{22}, \dots, x_{2n}$) is shown in formula 10, where s_k is the standard deviation of the k th component in Figure 4.

$$\text{dis}(a, b) = a - b_{\text{normal}} = \sqrt{\sum_{k=1}^n \left(\frac{x_{1k} - x_{2k}}{s_k} \right)^2}. \quad (9)$$

At this point, we have completed the improvement of SGAN, and the loss function of its final generator G is

$$L_G = E_{\hat{x} \sim p(\hat{x})} [\log P_d(y = K + 1 | \hat{x})] + \lambda * [D(x, \hat{x}) + \text{dis}]. \quad (10)$$

The specific training process of the improved SGAN is in Table 1.

In this paper, the structure of the generator and discriminator of the improved semisupervised generation countermeasure network is selected as CNN, the size of convolution kernel in convolution layer and deconvolution layer is 3×3 , and the step size is 2. The results of each convolution and deconvolution are normalized in batches to avoid overfitting and improve the training speed of the model. The learning rate is set to 0.0001, and the cross-entropy loss function is selected as the model error function, which is helpful to improve the classification effect.

We also smooth the 0 and 1 data tags in the model to enhance the anti-interference ability of the network. However, in order to keep the optimal discriminant function of the discriminator unchanged and avoid the occurrence of pattern collapse, we keep the tags with a value of 0

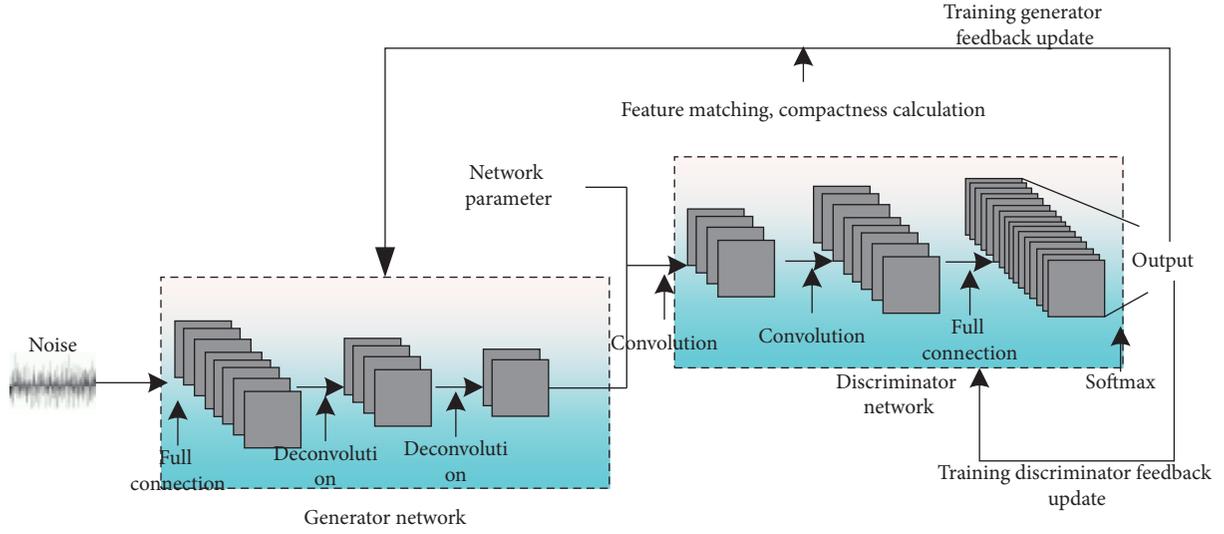


FIGURE 3: Fault diagnosis model based on improved generation countermeasure network.

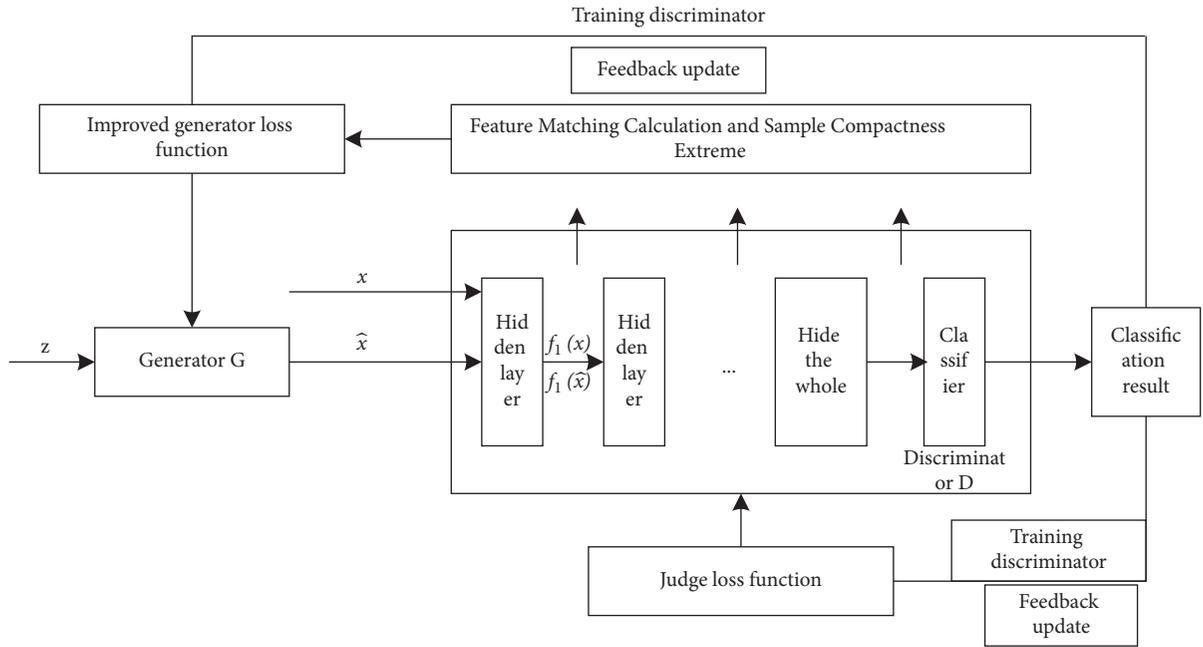


FIGURE 4: Algorithm flow chart of improved SGAN.

TABLE 1: Improved SGAN algorithm steps.

Algorithm: Improved SGAN model training based on small batch random gradient descent algorithm
 Inputs: $p(z)$: Random noise distribution, $p(x)$: True data distribution, λ : Specific gravity coefficient, m : Iteration times
 Output: Trained discriminator D

For $i=1$ to m do
 N Noise samples $\{Z^{(1)}, Z^{(2)}, \dots, Z^{(n)}\}$ subject to $P(z)$ distribution
 N labeled samples obeying $p(x)$ distribution $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}) \dots (x^{(n)}, y^{(n)})\}$
 N unlabeled samples following $p(x)$ distribution $\{\tilde{x}^{(1)}, \tilde{x}^{(2)}, \dots, \tilde{x}^{(n)}\}$
 Keeping the parameters of generator G unchanged, update the parameters of discriminator D according to the following formula

$$\min_{\theta(D)} \frac{1}{n} \sum_{i=1}^n \{-\log P_d(y^{(i)} | x^{(i)}, y^{(i)} < K + 1) - \log(1 - P_d(y = K + 1 | G(z^{(i)})))\}$$

 Keeping the parameters of discriminator D unchanged, update the parameters of generator G according to the following formula

$$\min_{\theta(G)} \frac{1}{n} \sum_{i=1}^n \{\log P_d(y = K + 1 | G(z^{(i)}))\}$$

 end

unchanged, and only change the tags with a value of 1 to 0.9; that is, smooth the unilateral tags.

In addition, in order to strengthen the generalization ability of the model, we also set a trigger condition for model modification for the trained discriminator model [15, 16]. Every time the discriminator diagnoses the network fault, the data that is wrongly diagnosed will be stored in the database. When the stored data is larger than one-third of the training data, the retraining of the model will be triggered. The training data consists of the original training data and the data for diagnosing errors, thus completing the correction of the discriminator model. The flow chart is as follows in Figure 5.

3. Generate a Fault Diagnosis Model Combining Countermeasure Network with Convolution Neural Network

Although the semisupervised fault diagnosis algorithm based on improved generation countermeasure network proposed in the previous section solves the problem that the diagnosis model can identify network faults autonomously under the condition of a small amount of marked data and a large amount of unmarked data, two constraint functions are added to the loss function of the generator network to stabilize and accelerate the convergence of the model to achieve Nash equilibrium. However, with the increase of the input network characteristic parameters, the convergence of the model will become more and more difficult, so we consider reducing the complexity of the discriminator network in the model to accelerate the convergence of the model.

In the improved semisupervised fault diagnosis model of generative countermeasure network, the discriminator mainly undertakes two tasks: one is to identify true and false data and the other is to classify the real data, which will undoubtedly increase the difficulty of convergence of the model. Therefore, the task of classifying the real data is extracted by the fault diagnosis model based on convolution neural network, and the discriminator only needs to identify the generated data and the real data [17, 18]. The disadvantage of convolution neural network discussed before is that it is a supervised learning method. It is not suitable for the scene with only a small amount of marked data, but now we can generate the data of each type of fault through the improved generation countermeasure network and then combine the generated data with the real data to train the fault diagnosis model based on convolution neural network to improve the accuracy of the diagnosis model [19, 20].

The fault diagnosis model proposed in this paper can be divided into two steps: data generation and fault diagnosis, as shown in Figure 6.

3.1. GAN Generates Network Data. Each node selects eight key network performance parameters, namely, RSRQ, DCR, HO, RSRP, ERAB, SNR_UL, SNR_DL, and LER, and obtains the data at time t as shown in

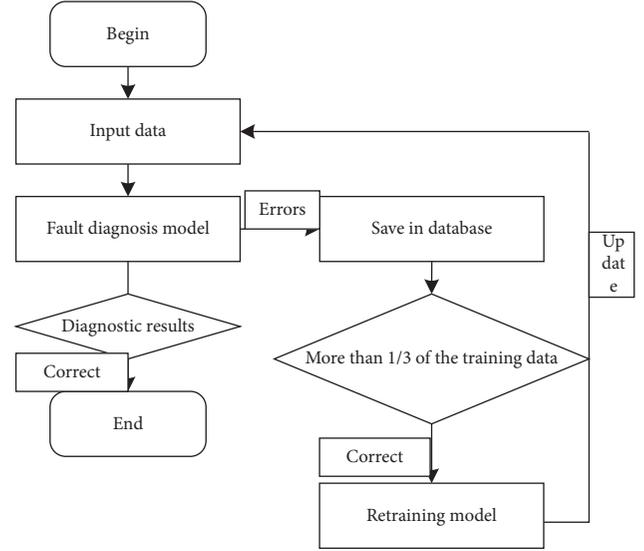


FIGURE 5: Flow chart of model correction.

$$\begin{aligned}
 X^t &= \begin{bmatrix} KPI_1^{t-T+1} & \cdots & KPI_1^t \\ \vdots & \ddots & \vdots \\ KPI_8^{t-T+1} & \cdots & KPI_8^t \end{bmatrix} \\
 &= \begin{bmatrix} \vdots \\ \frac{1}{T} \sum_{i=t-T+1}^t KPI_1^i \frac{1}{T} \sum_{i=t-T+1}^t KPI_2^i & \vdots \\ \frac{1}{T} \sum_{i=t-T+1}^t KPI_8^i \end{bmatrix} \quad (11) \\
 &= \begin{bmatrix} KPI_{avg-1}^t \\ KPI_{avg-2}^t \\ \vdots \\ KPI_{avg-t}^t \end{bmatrix}
 \end{aligned}$$

Considering that the network scenario in this paper is a dense heterogeneous wireless network, there is a certain correlation between base stations [21, 22], so we select the nearest 7 base stations as the neighbor base stations of the current base station. In order to ensure the same data format, when the number of neighbor base stations is less than 7, we use the average value of the neighbor base stations of the current base station to fill the insufficient data samples. Finally, the data of each base station at T time is shown in

$$X^t = \begin{bmatrix} X_{cur}^t \\ X_1^t \\ \vdots \\ X_7^t \end{bmatrix} = \begin{bmatrix} KPI_{cur_{avg1}}^t & \cdots & KPI_{cur_{avg8}}^t \\ \vdots & \ddots & \vdots \\ KPI_{7_{avg1}}^t & \cdots & KPI_{7_{avg8}}^t \end{bmatrix}, \quad (12)$$

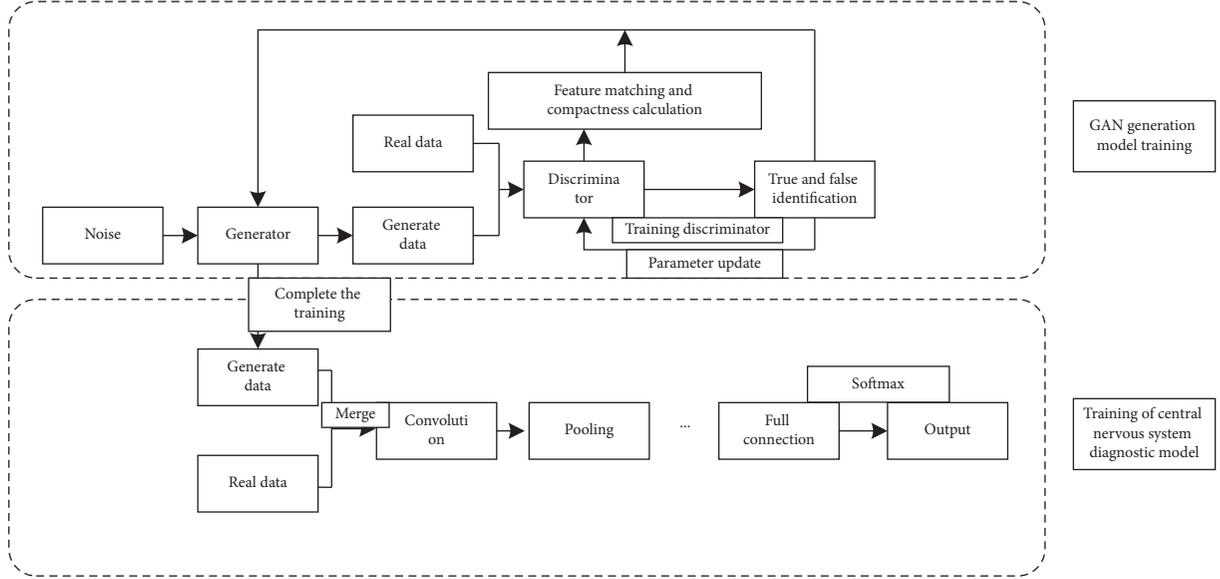


FIGURE 6: Generate a fault diagnosis model combining countermeasure network and convolution neural network.

where X_{cur}^t refers to the network key performance index parameter of the current base station at time t and X_i^t refers to the network key performance index parameter of the i -th neighbor base station at time t .

Assuming that there are K kinds of network faults, the labeled data samples are divided into K kinds according to the fault types, and then the classified data samples are input into the improved generation countermeasure network one by one for training, and the data of corresponding fault types are generated. The neural network structure for generating the countermeasure network still adopts the model parameters in the previous section, but the last two layers in the discriminator network are adjusted, the output format of the full connection layer is changed from [None, 7] to [None, 1], and the Softmax layer is also changed to sigmoid activation layer to ensure that the output results are in the range of 0~1, and the loss function of the network is changed accordingly, as shown in

$$L_G = E_{z \sim p(x)} [\log(1 - D(\hat{x}))] + \lambda * \left[\frac{1}{l} \sum_{i=1}^l E_{x \sim p(x)} f_i(x) - E_{z \sim p(x)} f_i(\hat{x}) \right], \quad (13)$$

$$L_D = - \left\{ E_{z \sim p(x)} [\log(D(x))] + E_{z \sim p(\hat{x})} [\log(1 - D(\hat{x}))] \right\}, \quad (14)$$

where input is the specific gravity coefficient, used to adjust the proportion between the cross-entropy loss function of the original generator network and the new target constraint function, X is the real data, \hat{x} is the generated data, x_r and x_f are the samples of the small batch real data set and the small batch generated data set respectively, n is the size of the small batch data set, l is the number of hidden layers in the discriminator, $f_i(\cdot)$ is the output of the hidden layer of the i layer, and $D(\cdot)$ is the output of the discriminator.

3.2. Network Fault Diagnosis Model Based on CNN. After all kinds of fault data are generated, we add corresponding fault labels manually and then merge them with real data as the training data set of convolution neural network.

The network structure of CNN diagnostic model in this section is shown in Table 2 where the convolution kernel size is 3×3 , the convolution step is 1, the pool core size is 2×2 , the pooling step size is 2, and the parameter of dropout operation in the full connection layer is 0.5; that is, half of the neurons in the full connection layer are temporarily discarded from the network during each training, so that the network structure of each training is different to avoid overfitting the model. The error function of the model still adopts the cross-entropy loss function, and the learning rate of the optimizer is set to 0.0001.

Specific fault diagnosis steps are as follows:

- (A) Building and generating a countermeasure network model according to the neural network parameters defined in (1) and dividing the marked real data into training data and test data according to a certain proportion.
- (B) Inputting the training data into the generated countermeasure network model according to the classification of fault categories, generating data of corresponding fault categories through the generator network after the model training converges, adding fault labels, and then combining the training data and the generated data as new training data of the convolution neural network diagnostic model.
- (C) Setting up a CNN model according to the neural network parameters defined in (2) and training the CNN model by using new training data.
- (D) Entering data from the test dataset to validate the performance of the diagnostic model.

TABLE 2: Network structure parameters of CNN fault diagnosis model.

Layer (operation)	Input format	Output format
Convolution layer 1	[None,8,8,1]	[None,8,8,32]
Batch normalization	[None,8,8,32]	[None,8,8,32]
Nonlinear activation	[None,8,8,32]	[None,8,8,32]
Pool layer	[None,8,8,32]	[None,4,4,32]
Convolution layer 2	[None,4,4,32]	[None,4,4,64]
Batch normalization	[None,4,4,64]	[None,4,4,64]
Nonlinear activation	[None,4,4,64]	[None,4,4,64]
Pool layer 2	[None,4,4,64]	[None,2,2,64]
Fully connected layer	[None,256]	[None,6]
Softmax	[None,6]	[None,6]

4. Performance Evaluation and Results

In this paper, GAN model and CNN model are based on TensorFlow framework, software using Pycharm development tools, hardware using Intel (R) Core (TM) i7-6700 CPU @ 3.40 GHz, memory 8G, operating system Windows 10 desktop in Table 3.

Network operation data is generated by OPNET software simulation; the hardware of OPNET network simulation software is Intel (R) Xeon (R) CPUE7-4870 v2 @ 2.3 GHz (2 processors), with 3G memory and Windows 7 operating system; a cellular network composed of 3 macro base stations and 15 micro base stations is built by using this simulation software. The overall coverage area of the network is $5\text{ km} \times 5\text{ km}$, and the coverage radius of each macro base station is 1 km. There are 5 micro base stations distributed within each macro base station, and users are randomly distributed in their respective cells. The specific parameters are shown in Table 3.

In network simulation, this paper mainly sets up five different kinds of faults, which are uplink interference F1, downlink interference F2, coverage fault F3, base station fault F4 and link fault F5, and fault-free state F0. Before the start of the simulation, we will set the occurrence and recovery time of these faults in advance, so as to manually add fault labels to the data generated by the simulation. The simulation setting time is 24 hours, the total duration of each fault is 2 hours, and each fault lasts for 30 minutes and then recovers. Finally, the network simulation generated a total of 29,160 pieces of data, which were normalized and preprocessed by neighboring base stations to obtain 1,620 pieces of data as shown in formula (9), which were divided into training data and test data in a ratio of 7:3, and the ratio of various faults in each data set was ensured to be consistent. Then, the fault labels of about 1/10 of the training data are retained, and the rest of the training data are deleted. Finally, three data sets are formed, namely, 120 training data sets with fault labels, 1014 training data sets without fault labels, and 486 test data sets, and the proportion of various faults in each data set is consistent. Finally, these data are input into the improved generative countermeasure network diagnosis model proposed in this paper for training and testing, and the simulation results are introduced in detail.

4.1. Improved GAN Diagnosis Model Influence of Different Batch Data Size on the Accuracy of Model Fault Diagnosis Firstly. Some nouns in this simulation are explained. Here, fault diagnosis accuracy refers to the percentage of correct diagnosis information to the total amount of diagnosis information, iteration times refer to the times of inputting complete training data sets into the model, and batch data size refers to the number of samples for training and updating model parameters once. In the actual neural network training, when the data set is large, if all the data is entered into the model at one time, it will affect the efficiency and optimization degree of model training, so we batch the training data set according to the batch data size to improve the convergence speed and performance of the model. Moreover, the size of small batch data set is also related to the batch data size in the improved calculation of compactness in this paper.

Figure 7 is the change of fault diagnosis accuracy under different batch data sizes. It can be seen from the figure that when the batch data length is too small, the fluctuation of diagnostic accuracy of the model is relatively large, because the number of samples updated each time is small, and the model is difficult to converge. If the batch data length is too large, although the fluctuation of diagnostic accuracy of the model is stable, the rise of diagnostic accuracy is relatively slow, which requires more iterations. Therefore, the final batch data size set in this article is 100.

4.2. The Influence of Specific Gravity Coefficient λ on the Fault Diagnosis Accuracy of the Model. In this paper, two constraint functions are added to the loss function of the generator network that generates the countermeasure network, and the selection of the proportion between the constraint function and the original loss function also has certain influence on the performance of the model, as shown in Figure 8. As can be seen from the figure, when the specific gravity coefficient is 0.1, the constraint function has no constraint on the total loss function; therefore, the convergence of the generator network model is difficult, and the accuracy of fault diagnosis fluctuates greatly. With the increase of the proportion of constraint function, the accuracy of fault diagnosis gradually rises slowly. When the specific gravity coefficient is 1, the fluctuation of the model begins to increase again. This is mainly because the output of the hidden layer in the middle of neural network is too emphasized, so this paper sets the specific gravity coefficient λ to 0.5 according to the simulation results.

4.3. Comparison of Fault Diagnosis Accuracy of Different Improved Generated Countermeasure Networks. Figure 9 is a diagram showing the variation of fault diagnosis accuracy of the generated countermeasure network model based on different improved methods. SDGAN refers to the semi-supervised generation countermeasure network model which improves the generator network and discriminator network into convolution neural network. SDGAN-F means that the loss function of the generator is improved on the basis of SDGAN, and the constraint function $D(x, \hat{x})$ based

TABLE 3: Simulation parameters of two-layer heterogeneous wireless network.

Simulation parameters	Macro base station	Micro base station
Number of base stations	1	3
Number of users	20/base station	10/base station
Transmission power	46 dBm	30 dBm
Standard deviation of shadow fading	8 dB	10 dB
Transmission loss model	SuburbanMacrocell (3GPP)/Free space	Indoor office environment/outdoor to indoor and pedestrian environment
Antenna gain	15 dBi	8 dBi
Operation mode	LTE 5 MHz FDD	LTE 10 MHz FDD
Receiving sensitivity	-110 dBm	-107 dBm
Base station selection strategy		Best suitable eNodeB
User distribution		Random distribution

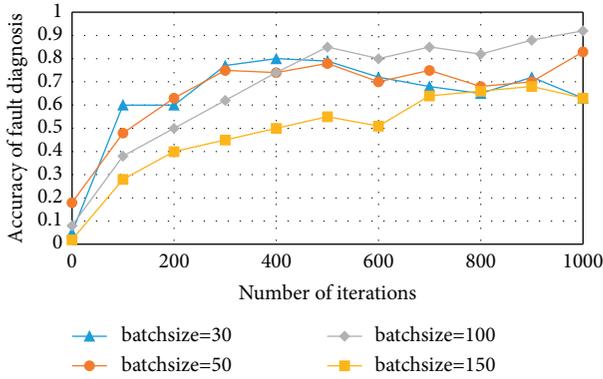


FIGURE 7: Variation of diagnostic accuracy under different batch data sizes.

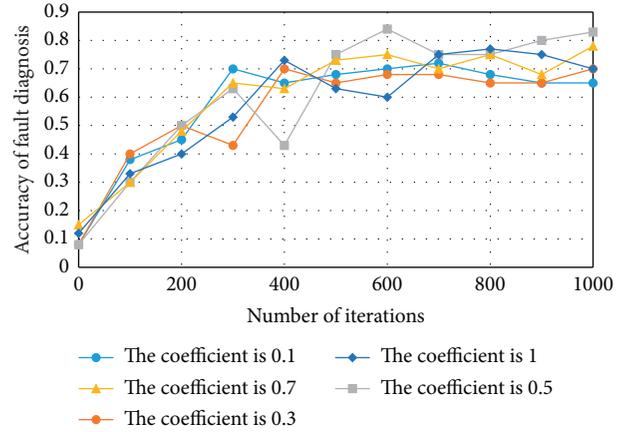


FIGURE 9: Variation of fault diagnosis accuracy based on different improved methods.

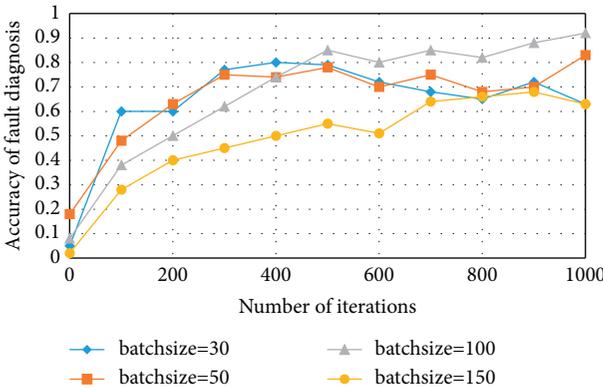


FIGURE 8: Variation of fault diagnosis accuracy under different specific gravity coefficients.

on feature matching is added. SDGAN-FM means that the loss function of the generator is modified on the basis of SDGAN, and a constraint function combining feature matching and batch sample compactness is added, such as formula (7). It can be seen from the figure that, compared with SDGAN, SDGAN-F does stabilize the convergence fluctuation of the generated countermeasure network in the training process because of adding the constraint function of feature matching, but compared with SDGAN-FM proposed in this paper, its fault diagnosis accuracy is still poor. The

final fault diagnosis accuracy of SDGAN-FM algorithm can reach 91.2%, while that of SDGAN-F algorithm can only reach about 70%. This is because the increased batch sample compactness constraint function can avoid the generator only generating data of a single fault category and improve the performance of the generator, thus improving the fault diagnosis accuracy of the discriminator.

4.4. *The Influence of the Amount of Data Generated by the Generated Countermeasure Network on the Fault Diagnosis Accuracy of the Model.* In this paper, a fault diagnosis model combining generated countermeasure network with convolution neural network is proposed; its data sets are 120 training data sets with fault labels and 486 test data sets. After classifying the training data sets according to fault categories, they are input to generate data of corresponding categories in the generated countermeasure network. After 1000 iterations, the loss functions of the generator and discriminator tend to be stable and the model converges.

Then, the generated data and the original training data are combined and input into the convolution neural network to train the fault diagnosis model. Table 4 compares the fault diagnosis accuracy of the models under different generated data quantities. As can be seen from Table 4, with the increase of generated data, the fault diagnosis accuracy of the

TABLE 4: Fault diagnosis accuracy of the model under different generated data quantities.

Generated data amount/original data amount	1	2	3	4	5
Accuracy of fault diagnosis	0.827	0.919	0.933	0.986	0.988

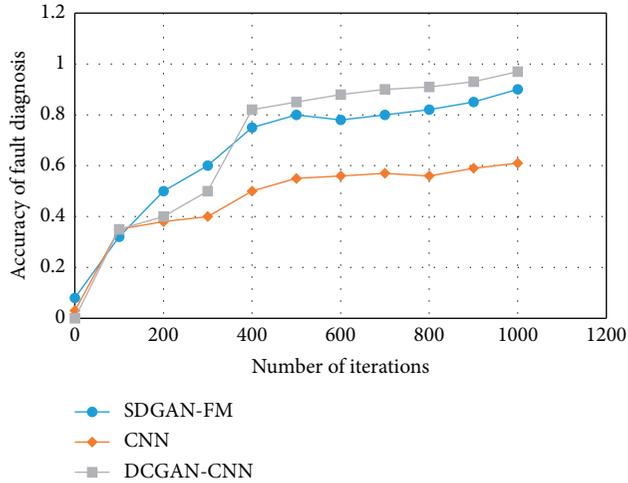


FIGURE 10: Comparison of fault diagnosis accuracy of different algorithms.

model is gradually improved. However, when the amount of generated data is 4 times the original training data, the improvement of fault diagnosis accuracy of the model will be relatively small if the amount of generated data is increased, and it takes more time to generate more data. Therefore, considering the overall performance of the model, this paper sets the amount of data generated by the generated countermeasure network to be 4 times the original training data.

4.5. Comparison of Fault Diagnosis Accuracy of Different Algorithms. Figure 10 is a comparison of the fault diagnosis accuracy of SDGAN-FM, CNN, and DCGAN-CNN algorithms. Among them, SDGAN-FM is a semisupervised fault diagnosis algorithm based on improved generation countermeasure network proposed in this paper, the training dataset includes a labeled dataset and an unlabeled dataset, CNN is a fault diagnosis algorithm based on convolution neural network, and the training data set only contains tagged data sets. DCGAN-CNN is a fault diagnosis algorithm based on the combination of generated countermeasure network and convolution neural network, and the training data set only contains tagged data sets. It can be seen from the figure that the final fault diagnosis accuracy of CNN algorithm is low due to the small amount of labeled data, while SDGAN-FM algorithm stabilizes the convergence fluctuation of the generated countermeasure network and makes full use of a large number of unlabeled data set, but the final diagnosis accuracy is only 91.2% due to the high complexity of the model. DCGAN-CNN algorithm makes full use of the characteristics of generated countermeasure network which is good at simulating data distribution and the good classification ability of convolution neural network,

makes up for the shortcoming of insufficient training data set of convolution neural network with generated countermeasure network, and finally obtains better fault diagnosis accuracy, which can reach 98.6%.

5. Conclusion

With the advent of the era of big data and the rapid development of technologies such as deep learning, computers with powerful computing power are needed to mine and extract key information from massive data by using complex neural network models. This paper presents a semi-supervised antinetwork fault diagnosis algorithm based on improved generation, The algorithm effectively guarantees the convergence of the generated network model, makes full use of a large number of trouble-free label data sets, and obtains better fault diagnosis accuracy. Then the diagnosis model is further optimized. The task of fault classification is completed by convolution neural network. In order to simplify the discriminant function of the network, the generated inverse network is only responsible for generating fault samples. The simulation results also show that the fault diagnosis algorithm based on generated inverse network combined with convolution neural network achieves better fault diagnosis accuracy and saves the overhead of manually marking a large number of data samples. In the next step, the performance comparison of this method under different network models is further studied. At present, the learning process of GAN can have a collapse problem, where the generator starts to degenerate and always produces the same sample points, unable to continue learning. When the generated model crashes, the discriminant model will also point to similar sample points in similar directions, and the training cannot continue, and this is also one of our research directions in the future.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding this work.

Acknowledgments

This work was supported by the research project of Taishan University: “ Intelligent monitoring system for pedestrian detection ” (No. 2020HX040), 2020.10-2022.12.

References

- [1] J. Gu, M. Li, L. Yu, S. Li, and K. Long, "Analysis on link travel time estimation considering time headway based on urban road RFID data," *Journal of Advanced Transportation*, vol. 2021, no. 3, 19 pages, Article ID 8876626, 2021.
- [2] J. Zhang and J. Chen, "An adaptive clustering algorithm for dynamic heterogeneous wireless sensor networks," *Wireless Networks*, vol. 25, no. 8, pp. 455–470, 2019.
- [3] L. Cheng, Y. Yang, K. Zhao, and Z. Gao, "Research and improvement of TF-IDF algorithm based on information theory," *Advances in Intelligent Systems and Computing*, vol. 905, pp. 608–616, 2020.
- [4] R. Yu, M. Lyu, J. Lu, Y. Yang, G. Shen, and F. Li, "Spatial coordinates correction based on multi-sensor low-altitude remote sensing image registration for monitoring forest dynamics," *IEEE Access*, vol. 8, p. 1, 2020.
- [5] X. Dai, X. Wu, B. Wang, and L. Zhang, "Semisupervised scene classification for remote sensing images: a method based on convolutional neural networks and ensemble learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 6, pp. 869–873, 2019.
- [6] M. Wang, Z. Zhang, K. Li, Z. Zhang, Y. Sheng, and S. Liu, "Research on key technologies of fault diagnosis and early warning for high-end equipment based on intelligent manufacturing and Internet of Things," *The International Journal of Advanced Manufacturing Technology*, vol. 107, no. 3, pp. 1039–1048, 2020.
- [7] L. Wang, H. Tian, and H. Zhang, "Soft fault diagnosis of analog circuits based on semi-supervised support vector machine[J]," *Analog Integrated Circuits and Signal Processing*, vol. 108, no. 3, pp. 1–11, 2021.
- [8] Y. Wu, R. Zhao, W. Jin, T. He, S. Ma, and M. Shi, "Intelligent fault diagnosis of rolling bearings using a semi-supervised convolutional neural network," *Applied Intelligence*, vol. 51, no. 4, pp. 2144–2160, 2021.
- [9] C. Fan, X. Liu, P. Xue, and J. Wang, "Statistical characterization of semi-supervised neural networks for fault detection and diagnosis of air handling units," *Energy and Buildings*, vol. 234, no. 12, Article ID 110733, 2021.
- [10] S. Shao, P. Wang, and R. Yan, "Generative adversarial networks for data augmentation in machine fault diagnosis," *Computers in Industry*, vol. 106, pp. 85–93, 2019.
- [11] Z. Q. Liu, T. Zhang, and Y. F. Wang, "Research on local dynamic path planning method for intelligent vehicle lane-changing," *Journal of Advanced Transportation*, vol. 2019, Article ID 4762658, 10 pages, 2019.
- [12] C. Jiang, J. Wan, and H. Abbas, "An edge computing node deployment method based on improved k-means clustering algorithm for smart manufacturing," *IEEE Systems Journal*, vol. 15, no. 2, pp. 2230–2240, 2021.
- [13] H. Zhao, B. Yang, L. Cao, and H. Li, "Data-driven enhancement of blurry retinal images via generative adversarial networks," in *Lecture Notes in Computer Science*, pp. 75–83, Springer, Cham, Switzerland, 2019.
- [14] B. Ssa, A. Hl, and C. Plb, "CEGAN: classification enhancement generative adversarial networks for unraveling data imbalance problems," *Neural Networks*, vol. 133, pp. 69–86, 2021.
- [15] D. Sun, J. Xu, H. Wen, and Y. Wang, "An optimized random forest model and its generalization ability in landslide susceptibility mapping: application in two areas of three gorges reservoir, China," *Journal of Earth Sciences*, vol. 31, no. 6, pp. 1068–1086, 2020.
- [16] Z. Liu, W. Zhan, J. Lai et al., "Balancing prediction accuracy and generalization ability: a hybrid framework for modelling the annual dynamics of satellite-derived land surface temperatures," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 151, no. MAY, pp. 189–206, 2019.
- [17] T. Du, H. Zhang, and L. Wang, "Analogue circuit fault diagnosis based on convolution neural network," *Electronics Letters*, vol. 55, no. 24, pp. 1277–1279, 2019.
- [18] S. Chen, H. Ge, H. Li, Y. Sun, and X. Qian, "Hierarchical deep convolution neural networks based on transfer learning for transformer rectifier unit fault diagnosis," *Measurement*, vol. 167, Article ID 108257, 2021.
- [19] X. Zhang, G. Chen, T. Hao, and Z. He, "Rolling bearing fault convolutional neural network diagnosis method based on casing signal," *Journal of Mechanical Science and Technology*, vol. 34, no. 2, pp. pp2307–2316, 2020.
- [20] S. Gao, Z. Pei, Y. Zhang, and T. Li, "Bearing fault diagnosis based on adaptive convolutional neural network with Nesterov momentum," *IEEE Sensors Journal*, vol. 21, no. 7, pp. 9268–9276, 2021.
- [21] A. Oa, A. Gi, and A. Ma, "Energy efficiency techniques in ultra-dense wireless heterogeneous networks: an overview and outlook," *Engineering Science and Technology, an International Journal*, vol. 23, no. 6, pp. 1308–1326, 2020.
- [22] R. Ahmad, E. A. Sundararajan, and A. Khalifeh, "A survey on femtocell handover management in dense heterogeneous 5G networks," *Telecommunication Systems*, vol. 75, no. 4, pp. 481–507, 2020.