ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20230501000593
Original scientific paper

Fault Diagnosis of Transfer Learning Equipment Based on Cloud Edge Collaboration + Confrontation Network

Ping ZOU*, Zhenji ZHANG, Lei JIANG

Abstract: With the continuous improvement of product quality, production efficiency, and complexity, higher requirements are put forward for the reliability and stability of equipment, and the difficulty of real-time diagnosis of faults and functional failures is also increasing. The traditional fault diagnosis methods based on signal processing and Convolutional neural network cannot meet the requirements of on-site online real-time fault diagnosis of equipment. One is that the vibration signals on the industrial site are superimposed on each other, nonlinear and unstable and traditional feature extraction methods take a long time, resulting in unstable extraction results. Second, massive data and fault diagnosis algorithms need rich computing and storage resources. The traditional Convolutional neural network method conflicts with the real-time response requirements of fault diagnosis. At the same time, different models of fault diagnosis models have poor generalization ability, and the diagnostic accuracy is not high or even impossible to diagnose. To solve the above problems, this paper proposes a fault diagnosis method based on industrial Internet platform, which is equipment cloud edge collaboration + adaptive countermeasure network Transfer learning. On the edge side, the vibration signals collected from key components of the model are processed using empirical mode decomposition (EEMD) to solve the problem of signal nonlinearity and stationarity. In the cloud, EEMD signals of different models are decomposed into source domain and target domain for confrontation training, which is used as the input of the improved domain adversarial network model DANN (Domain Adversarial Neural Networks), so as to improve the accuracy of fault diagnosis of different models by using cloud computing power and the improved adversarial network Transfer learning algorithm. Through the analysis of experimental data, this paper verifies that the model after the confrontation network Transfer learning is more accurate than the traditional fault diagnosis method. Through the coordination of computing resources and real-time requirements, real-time diagnosis of cloud side collaborative bearing fault is realized.

Key words: cloud edge; cooperative countermeasure; fault diagnosis; network Transfer learning

1 INTRODUCTION

During the operation of complex equipment, there are differences in the environment where different types of equipment are located, with overlapping edge signals and nonlinear instability. The trained diagnostic model is difficult to apply to cross terminal diagnosis of multiple types of equipment, and lacks effective data and model collaboration and adaptation mechanisms. A more coordinated dynamic and fast diagnostic method is needed to accurately diagnose faults between different types in real time.

The traditional fault diagnosis method using Fourier transform can collect data for real-time analysis at the edge, but due to the nonlinear and non-stationary characteristics of equipment vibration signals, it is difficult to achieve good analysis results. The machine learning methods represented by CNN (Convolutional Neural Networks) can automatically train and extract features, solve the bottleneck problem of knowledge acquisition, and have high diagnostic accuracy. The main drawbacks are that on the one hand, they need to be retrained for similar models, and the diagnostic model is difficult to apply to cross terminal diagnosis of multiple model devices. On the other hand, the model requires sufficient training with a large amount of labeled training data to complete accurate fault identification, and the collection of labeled samples and accurate training of the model require a large amount of computational time and resources. Therefore, at present, similar deep learning tasks are mostly conducted in the cloud with abundant computing and storage resources.

With the development of Internet of Things technology, more and more data is generated and processed at the edge of factories. Processing some data at the network edge will be more efficient. However, on the one hand, training and diagnosis through the edge is limited by the computing and storage capabilities of the edge, resulting in an increase in training time. On the other hand, it is also difficult to collect a large amount of data that can meet the needs of sufficient model training. Therefore, this paper proposes a cloud edge collaboration + adaptive countermeasure network Transfer learning fault diagnosis method. At the edge side, the vibration signals collected by key components of the model are processed by empirical mode, and the signal source domain and target domain are trained by confrontation at the platform side. Finally, the migration model after training is used for online fault diagnosis of similar models to solve the problem of online rapid diagnosis of equipment under different working conditions, thus improving the accuracy of equipment fault diagnosis. This article verifies through experiments that the above methods have higher fault diagnosis accuracy, and also verifies the feasibility of the cloud edge collaboration + adversarial network method proposed in this article for equipment online fault diagnosis.

2 LITERATURE REVIEW

2.1 Development of Cloud Edge Collaboration Technology

Cloud edge collaboration based on cloud computing and edge computing is the most important feature of the industrial Internet. Cloud edge collaboration can achieve good fit and efficient matching for different equipment health management scenarios, and can maximize the value of edge side and cloud data processing and analysis. Cloud computing technology can be traced back to 1956, when Christopher Strachey officially proposed the concept of virtualization [1], which is the foundation of cloud computing development. With the development of network technology, cloud computing has gone through the formation period from 2006 to 2010, the development period from 2010 to 2015, and the application period from 2015 to 2020. It has now entered a mature period and become the core of enterprise digital priority strategy. Edge computing can be traced back to the Content Delivery Network (CDN) proposed by Akamai in 1998. It relies on cache servers deployed in different places to point users' access to the nearest cache server through load balancing, content distribution, scheduling and other functional modules of the central platform.

Since the development of the concept of cloud side collaboration, the global management capability has been gradually improved to promote the distributed development of computing resources. By building a cloud side collaboration management platform in the cloud, edge computing nodes are managed uniformly, and the collaboration between cloud side and edge computing nodes is realized from resources, data, services, applications, models, etc., to promote the distributed development of computing resources. Collaborative processing and analysis of cloud edge data effectively improves data utilization efficiency. With the advantages of convenient edge data collection and real-time processing and calculation, artificial intelligence models such as fault diagnosis gradually sink from the central cloud to the edge. By conducting collaborative reasoning and training on the edge and cloud, the problem of industrial intelligence diagnosis being applied in enterprise applications is solved. MA Xue, WEN Chenglin [2] and others proposed an asynchronous quasi cloud/edge/client collaborative joint learning mechanism for fault diagnosis, established a new asynchronous quasi cloud/edge/client collaborative Federated learning mechanism, and verified the effectiveness of the algorithm through the data of rotating machinery. Sun Ming [3] proposed a new task unloading strategy based on Deep reinforcement learning CTOSDRL to solve the problem of task unloading for multi-user collaboration in the cloud.

2.2 Development of Fault Diagnosis Based on Deep Learning

In recent years, with the development of artificial intelligence, fault algorithms based on deep learning have gradually been used for industrial field fault diagnosis. The fault diagnosis method represented by Convolutional neural network does not need complex signal analysis technology to extract fault features, and realizes end-to-end automatic extraction of fault features and fault diagnosis.

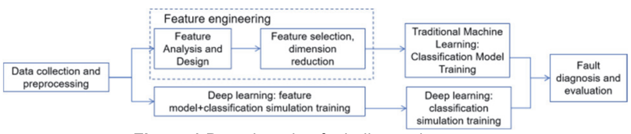


Figure 1 Deep learning fault diagnosis process

Despite the rapid development of deep learning in the field of fault diagnosis, there are still the following problems: firstly, data models are difficult to adapt under different operating conditions, and training data and test data need to meet the same probability distribution. The second is that a large amount of labeled data is required for training, and the accuracy of fault prediction is not high under the condition of a small number of samples. The two assumptions are often not satisfied in fault diagnosis in industrial sites. Therefore, the bearing fault diagnosis method based on deep learning mainly faces the problem of insufficient fault sample size, insufficient network model training caused by imbalanced data, and decreased diagnostic accuracy across working conditions.

Domestic and foreign scholars have conducted some research and innovation on these two issues. Yoshinski et al. [4] published a study on the transferability of deep neural networks in NIPS to address the issue of fault diagnosis across operating conditions due to insufficient sample size. They proposed that the transferability between basic tasks and target tasks will decrease as the difference between the two increases, but transferring relevant knowledge from similar tasks can help achieve the training of target tasks. In recent years, Transfer learning has gradually attracted the attention of the academic community. The concept of Transfer learning TL (Transfer Learning) [5] has been studied as early as the 1980s [6-8], which is used to improve the problem that machine learning methods have few learning training samples and low quality. Transfer learning uses a small number of high-quality samples to train high-precision models. It deals with different probability distributions by extracting common models under different conditions, and solves the problem of adaptability of equipment models and environmental difference models [[9]. The principle that Transfer learning can be realized is that Convolutional neural network has common features in shallow learning. In case of insufficient samples, Transfer learning can be used to transfer these common features from other trained networks, so as to save training time and obtain better recognition results [14]. Ganin et al. proposed a domain adversarial training of neural networks (DANN), successfully applied the idea of Generative adversarial network to Transfer learning, experimented in document emotion analysis and image classification, and achieved the most advanced domain adaptability on a standard basis [11].

Yu et al. proposed a new dynamic adversarial adaptation network (DAAN). The dynamic learning domain is invariant, and the relative importance of the global and Local field distribution is quantitatively evaluated to achieve cross domain classification tasks [12], 13].

To sum up, use the industrial Internet platform to carry out Transfer learning network confrontation training on fault diagnosis and predictive maintenance models, extract key features of different types of equipment at the edge side, use the edge Real-time computing advantages, carry out empirical mode analysis and preprocessing on equipment fault signals, build an effective data set containing vibration signals, and solve the problem of dynamic acquisition and processing of fault features of massive equipment data. In the cloud, the improved adaptive countermeasure network Transfer learning method is used for training to solve the online real-time diagnosis problem of multiple types of equipment under the same working condition and variable working condition, and the fault diagnosis accuracy is higher. Scholars at home and abroad have made relevant progress in this field. Zhang Yong [15] and others used the industrial Internet plus + Transfer learning technology to conduct real-time qualitative analysis (health status) and quantitative analysis (remaining service life) on the status of bearings and tools of key industrial components, to achieve health status assessment and degradation trend prediction of industrial equipment; Chen Jiaxian, Mao Wentao et al. proposed a deep temporal feature transfer based bearing residual life prediction method for predicting equipment life.

3 RESEARCH METHODOLOGY

3.1 Cloud Edge Collaboration Process

The advantages of the terminal mainly include two aspects. One is the rich computing resources, which can carry out massive data and complex large computing power operations, especially Big data and machine learning model training in complex scenarios. The second is to have sufficient storage resources, which can store massive training samples through distributed or centralized storage. But there are also drawbacks to cloud computing, such as poor sensitivity to low latency and high reliability calculations, poor task customization ability, and often inability to meet real-time requirements. Due to its proximity to application objects, the edge end has the characteristics of good real-time responsiveness and a single service object, allowing for personalized service customization. To solve the problem of real-time collaborative diagnosis of device fault data, this article combines the advantages of cloud and edge, and carries out real-time online diagnosis and analysis of device faults based on cloud edge collaboration. The overall design framework of cloud edge collaboration is shown in Fig. 2.

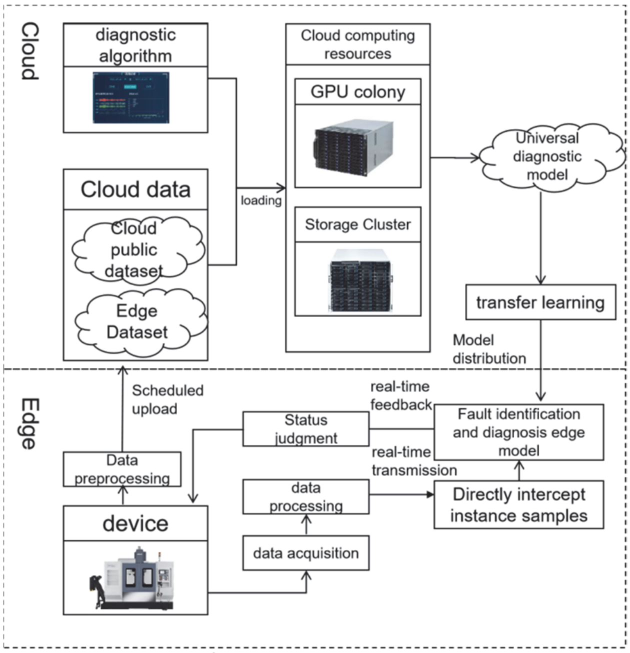


Figure 2 Cloud edge collaboration mechanism

Cloud: Mainly used for storing massive training samples and training general models. Among them, cloud data consists of two parts: one is the public dataset for fault diagnosis, and the other is the fault diagnosis sample dataset collected and uploaded by the edge side, which is preprocessed by the edge side. Secondly, based on the abundant training sample resources, storage resources, and computing resources in the cloud, the device fault diagnosis model is continuously trained and updated to obtain a universal training model that can serve different diagnostic scenarios as a generalization result. Finally, the trained model is distributed from the cloud to the edge.

Edge end: Collect real-time data under specific working conditions through the edge gateway, and on the one hand, transmit it to the cloud for storage and training. On the other hand, training samples are formed after edge processing to personalize and modify the universal diagnostic model migrated from the cloud to the edge, thereby improving the applicability of specific diagnostic tasks and forming a personalized model for real-time fault diagnosis.

From Fig. 2, it can be seen that there are two main methods for data interaction between cloud and edge in the bearing fault diagnosis task. Firstly, the edge will regularly upload personalized samples stored locally to the cloud for storage, enriching the cloud's training and validation sets. Secondly, as the dataset stored in the cloud is updated, the optimization of universal diagnostic model parameters is completed through training on a regular basis, and the optimized model parameters are transferred to the edge for model diagnosis.

3.2 Edge Side EEMD Data Preprocessing

Extracting fault features on the edge side is a crucial step in equipment online fault diagnosis, which directly affects the accuracy of diagnosis. Due to the severe superposition and coupling of vibration signals between components such as aerospace engines and high-speed train transmission systems, the vibration signals exhibit nonlinear and non-stationary characteristics. EEMD is an adaptive signal decomposition method that is more suitable for processing nonlinear and non-stationary signals. The combination of EEMD and entropy can effectively reflect the working conditions of components. Therefore, this article chooses to extract the EEMD entropy features of the fused information to form a feature vector.

3.2.1 Collective Empirical Mode EMD

The principle of EMD analytical decomposition is a self adaptive wavelet power signal modal analytical decomposition calculation and analysis method proposed by Huang [17]. Unlike other FFTs and wavelet transforms, EMD does not require manual selection of basic wave functions. It is a self adaptive wavelet signal modal decomposition method that can be applied to various characteristic modal signals based on the values of the power signal processing data itself, suitable for widely used decomposition processing of nonlinear, non-stationary, and non optimized power signals [18]. One of the main essential decomposition principles used in EMD decomposition to decompose different power signals is to perform various linear average unstable wavelet signal decomposition processes on the frequency bands of different power signals themselves, and transform the various adaptive mode signal decomposition wavelets of different power signals into wavelet transform functions under different power signal frequency bands, namely the Intrinsic Mode Function (IMF) of various characteristic mode decomposition wavelets. The specific steps of EMD formula decomposition are described as follows:

1) Identify $x(t)$ all maximum and minimum points of the signal;
2) The maximum points are $e_{\max}(t)$ fitted to the upper envelope and the minimum points to the lower envelope by cubic Spline interpolation interpolation function $e_{\min}(t)$;
3) Take the $e_{\min}(t)$ average value $m_1(t)$;
4) $x(t)$ Subtract to $m_1(t)$ obtain a new signal $h_1^1(t)$;
5) Determine $h_1^1(t)$ whether the two conditions of the IMF are met. If not, $h_1^1(t)$ repeat steps 1-4 as the new original signal until the $h_1^1(t)$ conditions of the IMF component are met. Reference [16] suggests that it is appropriate to determine the conditions for termination of screening as 0.2 to 0.3 times the standard deviation;
6) $x(t)$ Subtracting the $c_1(t)$ signal to obtain the remaining signal $r_1(t)$;

$$r_1(t) = x(t) - c_1(t) \tag{3-1}$$

$|r_n(t)|$ The decomposition process will $r_1(t)$ continue as a new original signal, repeating steps 1-6 until $r_n(t)$ there are no periodic components present or when it is small enough, the decomposition process will stop. The final signal $x(t)$ is decomposed into a set of IMF components and the r_n sum of the remainder:

$$x(t) = \sum_{i=1}^{n} c_i(t) + r_n \tag{3-2}$$

In the equation, $c_i(t)(i = 1, 2, ..., n)$ represents a series of IMF components and r_n represents the remaining terms. Take the vertical vibration acceleration information collected under the large gear bearing of the transmission system gearbox as an example for EMD decomposition. In order to see the effect of EMD decomposition more clearly, FFT transformation was performed on IMF1~IMF6, and the decomposition results are shown in Fig. 3.

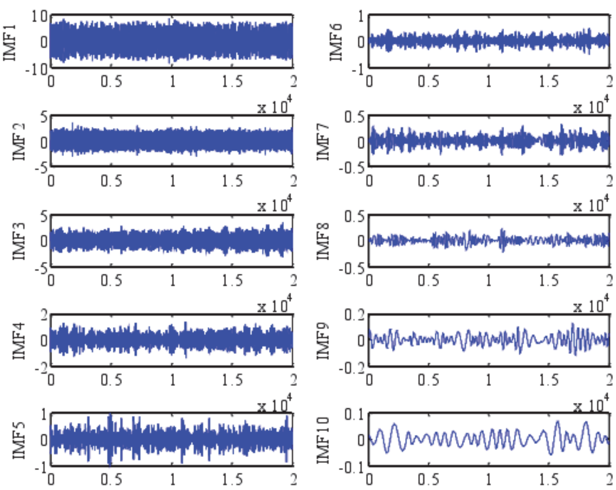


Figure 3 EMD decomposition results

From the frequency amplitude spectra of each IMF component in Fig. 4, it can be seen that the IMF3~IMF5 components exhibit frequency aliasing phenomenon.

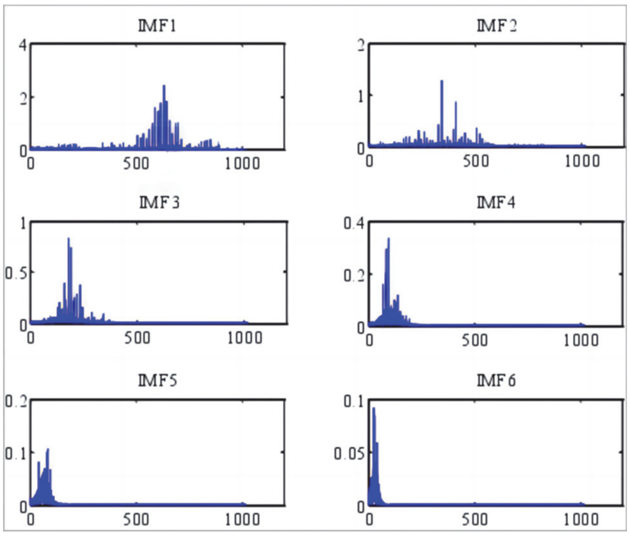


Figure 4 IMF1-IMF6 spectral map of EMD decomposition

3.2.2 Improved EEMD method

There are two main reasons for EMD frequency aliasing: (1) there is discontinuity in the signal; (2) there is interaction between signals. In order to continuously improve the EMD in this method, Wu and Huang [19] proposed the improved method EEMD. EEMD is essentially a power decomposition noise control analysis method for blank radio frequency noise control or auxiliary radio frequency signals. That is, we can directly decompose multiple blank radio frequency noise signals with only a certain signal frequency and amplitude value added by multiple control decompositions in a single radio frequency signal, so that the noise energy distribution of each signal extreme value and node in different power bands of a radio frequency signal is uniform. To solve the problem of modal aliasing caused by signal discontinuity, IMF components should be free from faults at various scales. In order to eliminate the impact of adding noise on the signal, the mean of the decomposed IMF components is taken as the final decomposition result. The specific steps of EEMD decomposition are as follows:

1) $x(t)$ Add white noise to the signal to $z(t)$ form a new signal $y(t)$;
2) $y(t)$ Perform EMD decomposition;
3) Repeat steps 1-2 N times, adding noise of different amplitudes each time;
4) Average the IMF decomposed N times;

$$\begin{cases} \overline{c_j} = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} c_{i,j} \\ \overline{r_j} = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} r_j \end{cases} (i = 1, 2, ..., N;\ j = 1, 2, ..., J) \tag{3-3}$$

In the formula, N is the number of aggregations; the j IMF component i obtained $c_{i,j}$ for the first decomposition; r_i is the i remaining term obtained through secondary decomposition; J is the number of IMF components, r_j is the average value of N factorizations. According to formula (4.10), the amplitude and aggregation number of white noise added have a significant impact on the decomposition results. In order to find all possible solutions during the screening process, the amplitude of adding white noise should be finite rather than infinitesimal. It is generally recommended to increase the amplitude of white noise to 0.1, 0.2, or 0.4 times the standard deviation. When the noise amplitude increases, the number of aggregations should also increase, and an increase in the number of aggregations will cause the calculation time to increase exponentially. Therefore, selecting appropriate decomposition parameters is the key to EEMD decomposition.

3.3 Cloud Based Deep Adversarial Adaptive Network DANN
3.3.1 Definition of Transfer Learning

Intelligent diagnosis of faults requires a large amount of labeled data, which is usually easy to obtain in the laboratory, but rare in actual working conditions. Traditional machine learning assumes that the training data and test data have the same distribution, and in actual working conditions, the distribution of training data and test data in a certain working condition is inconsistent, which leads to low accuracy of intelligent diagnosis of faults. In this case, the participation of Transfer learning greatly alleviates the contradiction between the requirements of intelligent diagnosis methods for labeled data and engineering practice, making it possible to carry out intelligent fault diagnosis even if the data distribution is inconsistent under the two working conditions.

Transfer learning [20] is a machine automatic learning method that can generalize the basic model obtained from machine learning in a research field to other fields. According to whether the source domain and target domain mark data, it can be divided into three categories: inductive Transfer learning, direct Transfer learning and unsupervised Transfer learning [21]. By reusing the marked data under different speeds and loads, the migrating fault diagnosis method improves the prediction effect of the fault diagnosis model under the target operation conditions without marks. She et al. [22] used the TrAdaBoost instance migration method for cross working condition fault diagnosis, while Xie et al. [23] used the Transfer Component Analysis (TCA) method to extract transferable features between domains. In the unsupervised Transfer learning domain adaptive fault diagnosis method, Lu et al. [24] proposed the use of deep neural networks for feature extraction, and the use of Maximum Mean Difference (MMD) to achieve the feature distribution alignment of source domain samples and target domain samples in the feature space; Han et al. [25] proposed a deep adversarial Convolutional neural network (DACNN) training cross condition fault diagnosis model.

The essence of Transfer learning algorithm is a learning process that uses the similarity between data, tasks or models to apply the models learned in the old field to the new field. Here $n+m_j=n+1$, the old domain is defined as the source domain ($D_s=\{x_i,y_x\}$ n_i =1), and the new domain is defined as the target domain (in this article, $D_t=\{x_j\}$ the actual operating data). The purpose of Transfer learning is y_i to learn the D_t knowledge of the target domain D_s (labels, y_j) with the help of the knowledge (labels). This method of using deep data network migration to automatically optimize learning is also called deep Transfer learning, in which the Loss function and its definition are as follows:

$$l=l_c(D_s,y_s)+\lambda l_{\mathrm{A}}(D_s,D_t) \tag{3-4}$$

Among them, l represents the final loss of the network; $l_c(D_s,y_s)$ indicates the loss of the network on labeled data; $l_{\mathrm{A}}(D_s,D_t)$ represents the adaptive loss of the network, reflecting the distribution difference between the source domain data and the target domain data. The goal of adaptation is to elevate the source domain and target domain data from low dimensional to high-dimensional space, minimizing the difference between the two distributions; λ is a weight parameter that balances the two parts.

3.3.2 Adaptive Network DANN

According to the game theory embodied in GANs [26-29], some scholars put forward the idea of adversarial Transfer learning. The central idea is to blur the boundaries between the source domain and the target domain in the confrontation, in order to achieve the goal of extracting common features and thereby achieve knowledge transfer. The adversarial transfer concept relies on the modules of the model and the game between them to achieve transfer, while minimizing a certain inter domain difference indicator is commonly used to achieve transfer. Therefore, on the basis of improving model performance by increasing depth, compared to rigid inter domain difference indicators, adversarial methods have a stronger abstraction ability in the process of learning common features.

In 2015, Ganin [30] and others proposed the DANN model. This model replaces the generator in GANs with a feature extractor, thereby transforming the original function of GANs to generate images into a feature extraction function, without changing the discriminator. In order to achieve classification tasks, a fully connected classifier is added separately after the extractor. In addition, Ganin et al. also put forward an idea called Gradient Reversal in DANN, which is used to optimize model parameters by backpropagation. After the discriminator is optimized by Gradient descent method, the discriminator is transferred to the extractor to reverse, so as to ensure that the purpose of confrontation training can be directly realized in the same cycle. The structure of DANN is shown in Fig. 5. Different from other Transfer learning methods, DANN focuses on embedding domain adaptively into the feature learning process, so that classification decisions are based on features with domain invariance or domain similarity. The trained model can predict the target domain without being affected by the differences in features between the two domains. Hu Ruohui et al. [31] applied DANN to cross domain recognition of bearing faults, integrating deep feature learning, domain adaptation, and label classification into a training process to minimize data feature differences under different operating conditions and achieve cross domain intelligent diagnosis of bearing fault modes.

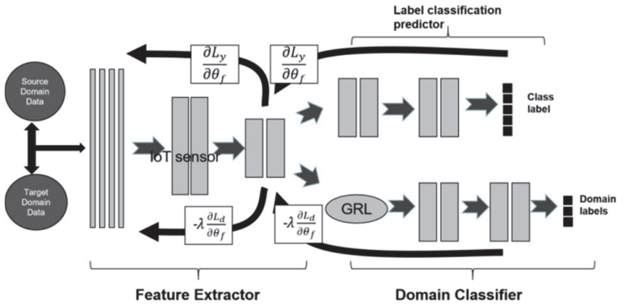


Figure 5 Domain adversarial network DANN structure framework diagram

The DANN structure mainly consists of three parts:

Feature Extractor: used to map and extract source domain data features into a specific feature category space. Using domain label feature predictor, users can accurately distinguish the values of all feature categories from all data in this source domain. However, domain feature discriminator cannot accurately distinguish which feature domain the source domain data comes from.

Label Predictor: classifies all data information from the data source domain to identify the correct label as much as possible.

Domain Classifier: identifies and classifies various data types in the entire feature data space, trying to distinguish which feature domain these data come from as much as possible.

The feature extractor and source domain label identification classifier together form a gradient feedforward neural network. After the feature extractor, we need to add a domain label discriminator, and in the middle, we connect through a Gradient Reverse Layer (GRL) that reverses along the gradient direction. Throughout the entire training process, the network continuously amplifies and minimizes the data loss (Loss) of the source domain label classification predictor for the loss of the entire labeled classification data from all source domains. The network continuously optimizes the data loss of the domain label discriminator to minimize the loss of all label data from the entire source domain and other target domains. The loss objective judgment function of DANN can be divided into two sub parts: 1. The loss term of the source domain label classifier. The classification of the source domain name by the source domain and the loss target is called the loss term.

1) Source domain classification loss term

For a m dimensional data point X, through a hidden layer G_f, the data point becomes a D dimension:

$$G_f(x;W,b) = \operatorname{sigm}(Wx+b) \tag{3-5}$$

Then, through a classification layer, the G_y classification results are obtained:

$$G_y(G_f(x);V,c) = \operatorname{soft\,max}(VG_f(x)+c) \tag{3-6}$$

The source domain classification loss for this point is defined as:

$$G_y(G_y(G_f(x_i)),y_i) = \log\frac{1}{G_y(G_f(x))y_i} \tag{3-7}$$

So the classification loss term of the source domain is defined as:

$$\min_{w,b,V,c}\left[\frac{1}{n}\sum_{i=1}^{n}L_y^i(W,b,V,c)+\lambda\cdot R(W,b)\right] \tag{3-8}$$

2) Domain classification loss term

When classifying domains, the label of the source domain is 0 and the label of the target domain is 1. Maximizing domain classification error means that the domain discriminator cannot distinguish between the source domain and the target domain, so that the source and target domains become aligned in distribution.

For any point from the source or target domain, the result obtained through the domain discrimination layer is:

$$G_d(G_f(x);u,z) = \operatorname{sigm}(u^T G_f(x)+z) \tag{3-9}$$

The domain classification error of this point is defined as:

$$\begin{aligned}L_d(G_d(G_f(x_i)),d_i) &= d_i\log\frac{1}{G_d(G_f(x_i))}+\\ &+(1-d_i)\log\frac{1}{1-G_d(G_f(x_i))}\end{aligned} \tag{3-10}$$

Cross entropy function: the $-y\lg\hat{y}-(1-y)\lg(1-\hat{y})$ form of the above formula is to put the minus sign into the logarithmic function. The domain classification error term is defined as:

$$\begin{aligned}&R(W,b)=\\ &\max_{u,z}\left[-\frac{1}{n}\sum_{i=1}^{n}L_d^i(W,b,u,z)-\frac{1}{n'}\sum_{i=n+1}^{N}L_d^i(W,b,u,z)\right]\end{aligned} \tag{3-11}$$

3) DANN objective function

The purpose of the DANN function is to minimize the error term in the source domain level classification and maximize the error term in the domain level classification. The target classification function is mainly used to minimize the classification problem. Therefore, a negative value symbol is added in front of the error term in the source domain level classification, and by introducing the λ value is used as a parameter to measure the weight value and balance value.

$$\begin{aligned}E(W,V,b,c,u,z) &= \frac{1}{n}\sum_{i=1}^{n}L_y^i(W,b,V,c)-\\ &-\lambda\left(\frac{1}{n}\sum_{i=1}^{n}L_d^i(W,b,u,z)+\frac{1}{n'}\sum_{i=n+1}^{N}L_d^i(W,b,u,z)\right)\end{aligned} \tag{3-12}$$

3.3.3 Improved DCGAN-DANN Method

DANN can perform adversarial learning on the source and target domains, provided that they require a large number of training samples. In actual working conditions, the sample size in the target domain is small, making it difficult to obtain enough samples for DANN training. In reality, there is a serious imbalance in the amount of bearing vibration data under different operating conditions, which makes it difficult to adapt to data features in different fields. Good fellows proposed a generative adversarial network, which randomly generates a large amount of label data required for training through algorithms. The generated data and real data have similar distribution characteristics [33]. Zhang et al. [32] proposed using a generative adversarial network to expand the data of bearing vibration signals. The model was validated on a small number of bearing signal datasets, and the generative adversarial network can effectively balance the difference in the number of samples between the source domain and the target domain.

To solve the problem of fault diagnosis for bearings under variable operating conditions with small target samples, this paper proposes an improved DANN method, DCGAN-DANN (Deep Convolutional Generative Adversarial Networks and Domain Adversarial Neural Networks) network, which uses the DCGAN method to generate samples of the target domain. DCGAN was proposed by Alec et al. and can simulate real data features to generate false data with the same distribution characteristics. The network consists of a generator (G) and a discriminator (D). As shown in Fig. 6, the generator is responsible for sampling the random vector Z in space as a vibration signal, and the discriminator is responsible for distinguishing between "true" and "false" signals. By extracting up sampled random vectors from different batch generators, a large number of pseudo samples serving as target domain signals are obtained.

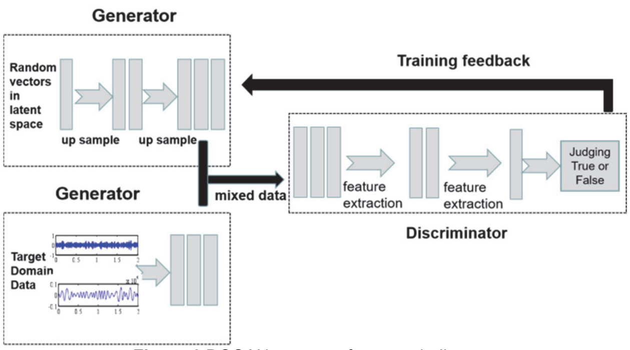


Figure 6 DCGAN structure framework diagram

After processing the generator pseudo samples, the generated signal and the real signal are passed into the discriminator. DCGAN requires distinguishing between the real and generated signals as much as possible, that is, maximizing the objective function $V(D, G)$. We label the generated signals with real labels and pass them into the training discriminator. The rise and fall of $V(D, G)$ form a confrontation. DCGAN generates more similar pseudo sample signal data by finding a balance between the source and target domains. The objective function that needs to be optimized is:

$$\min\max V(D,G) = E_{x\sim Pdata}\left[\log D(x)\right] + \\ + E_{x\sim PG}\left[\log\left\{1-D\left[G(Z)\right]\right\}\right] \tag{3-13}$$

In the above equation, $D(X)$ is the probability that the discriminator determines whether the real data is true, and $D(G(Z))$ is the probability that the discriminator determines whether the generated pseudo sample signal is true; P data is the true sample distribution, and PG is the prior distribution of vector Z.

After processing by DCGAN, the source and target domain signals are input into the DANN deep adversarial adaptive network. The implementation process is shown in Fig. 7. The DCGAN-DANN network uses the gradient propagation direction to realize the two domain adaptive processes of the adversary network migration diagnosis model. DCGAN and DANN complement each other in functionality. DCGAN provides a large amount of target domain sample data for cross domain feature transfer. DANN uses source domain data to characterize target domain features, making fault classification include a feature space of multiple operating conditions information.

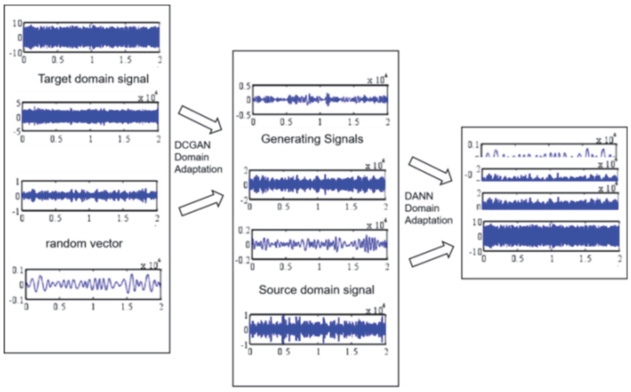


Figure 7 DCGAN-DANN data conduction diagram

The network first inputs DCGAN with a small number of labeled target domain signals and random vectors in space. In the gradient confrontation between the generator and discriminator, DCGAN maps the target domain signals and random vector features in space to the same space. The network mixes source domain label signals, labeled target domain pseudo sample signals, and small sample real target domain signals as inputs to the DANN network. The network maps generated signals and source domain signal features to the same space by confusing source domain and target domain labels, automatically adapting label features from different domains. The network domain adversarial adaptation and signal feature learning are carried out simultaneously, and the recognition of category labels will be affected by the influence of source domain labels, target domain pseudo labels, and domain labels.

3.4 Fault Diagnosis Based on Cloud Edge Collaboration + Confrontation Network Transfer Learning

3.4.1 Fault Diagnosis Framework

This section designs an equipment online fault diagnosis model based on cloud edge collaboration + Transfer learning, and its structure is shown in Fig. 8.

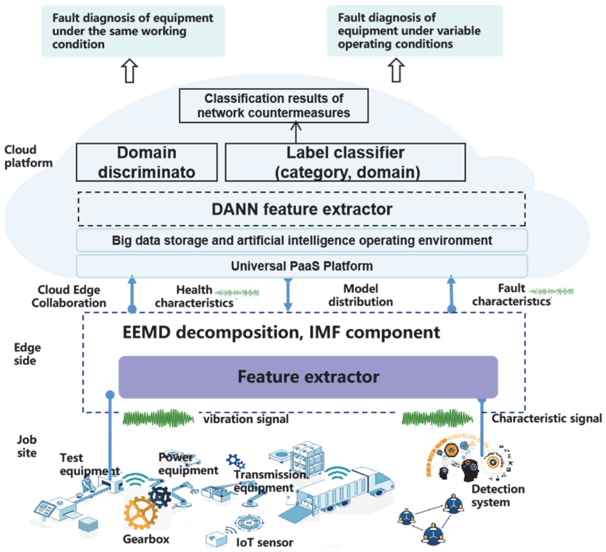


Figure 8 Design of cloud edge collaborative fault diagnosis model

Fault diagnosis under different working conditions is achieved through cloud edge collaboration + Transfer learning model:

On the edge side, perform EEMD decomposition on samples of different components, extract IMF fault feature components, and transfer the feature components IMF to the cloud.

In the cloud, calculating the classification loss of the source domain and the classification loss of the source and target domains, this model adopts a complete convolutional pooling structure combined with GAP strategy to replace the fully connected neural network[38][39] to construct classifiers and domain discriminators, and adopts the DCGAN-DANN network method to achieve stronger generalization performance.

3.4.2 Fault Diagnosis Process

First, the equipment health signal is acquired, the sample signal is decomposed by EEMD and Feature selection, and the DANN network countermeasure training is conducted in the cloud. The fault diagnosis process is shown in Fig. 9.

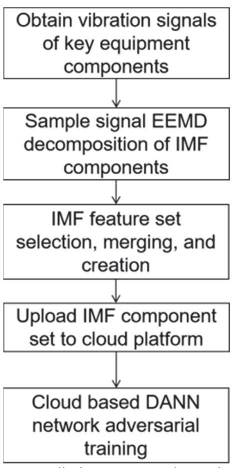


Figure 9 Equipment prediction process based on DCGAN-DANN

1) Acquisition of vibration signal data. Use all different states (normal and abnormal) of the device on the edge side as vibration training signals, and use them as the selected source domain vibration training signal data.

2) Sample signal EEMD decomposition. Extract fault features from the source domain signal of the vibration signal, perform EEMD decomposition, and use the decomposed bearing IMF component as input as the data source for DANN adversarial learning.

3) IMF feature set selection and creation. Based on the kurtosis of IMF components, select k IMF components with obvious fault feature information, merge them to create a dataset, and divide the dataset into training and testing sets.

4) Upload data to the cloud platform. Upload the processed IMF dataset sample to the cloud platform.

5) Cloud based DCGAN-DANN network adversarial training. Using the IMF dataset as the source of DANN network countermeasures data, model training is conducted on cloud platforms, and real-time fault diagnosis is performed.

3.4.3 DCGAN-DANN Algorithm Design

The cloud is based on the DCGAN-DANN method, which conducts adversarial training on the signal source and target domains after edge layer processing, reducing errors in the source and target domains, and improving the accuracy of fault diagnosis in the target domain. The fault diagnosis process based on deep transfer adversarial learning is shown in Fig. 10.

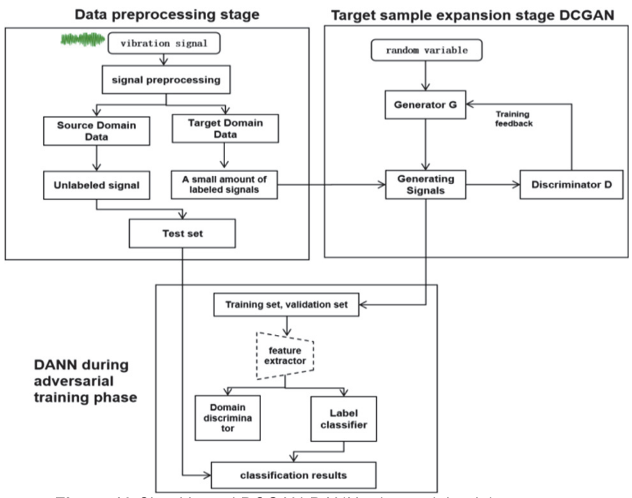


Figure 10 Cloud based DCGAN-DANN adversarial training process

The training process is divided into signal preprocessing stage, target sample expansion GCGAN stage, and adversarial training stage DANN.

1) Signal preprocessing stage: During training, the edge fault feature extraction signal is divided into source domain and target domain in the first stage, where the target domain maintains a small sample input to the DCGAN network.

2) Target sample expansion stage: First, input a small number of labeled target domain signals and random vectors in space into DCGAN, and continuously output labeled target domain pseudo signals. In the gradient confrontation between the generator and discriminator, DCGAN maps the target domain signal and the random vector features in space to the same space. The generated signal will inherit the labels of known signals based on the temporal characteristics of the target domain signal. The

network can generate target domain signals that are balanced with the number of samples in the source domain according to the needs of the model.

3) Adversarial training DANN stage: Mix source domain labeled signals, labeled target domain pseudo signals, and small sample target domain signals as inputs to the DANN network. DANN extracts the deep features of the source domain and target domain data training sets, and judges the source domain labels to obtain the source domain label Loss function Ls. The discriminator Loss function Lost is obtained from the discrimination between the target domain and the source domain, which is used to calculate the distance between the source domain and the target domain to minimize the distance between the two domains.

Table 1 DANN network adversarial model

Network model	Serial number	Network Description	Input	Output
Feature extraction network	1	First layer convolution	Shape = [50, 1, 7, 400] Batch_size = 50 Input_channels = 7	Output_channels = 16 Kernel_size = [1, 40]
	2	Second layer convolution	Input_channels = 16	Output_channels = 32 Kernel_ Size = [1, 40]
	3	Third layer convolution	Input_ Channels = 16	Output_ Channels = 32 Kernel_ Size = [1, 40]
Label Classful network	1	Fully connected layer	[64 * 1 * 15, 100]	BatchNorm1d (100) Relu (True) Dropout2d ()
	2	Fully connected layer	[100, 100]	BatchNorm1d (100) Relu (True)
	3	Fully connected layer	[100, 9]	Classifier = Softmax
Source domain and target domain Classful network	1	Fully connected layer	[64 * 1 * 15, 100]	BatchNorm1d (100) Relu (True)
	2	Fully connected layer	[100, 2]	Classifier = Softmax

In the process of training the multi state classification model, an MMD adaptation layer is added after the feature layer to extract the features of the source domain and the target domain and map them to a common space. According to the Loss function of the domain discriminator, the parameters are constantly updated, and after multiple iterations of optimization, the multi state classification model of different types of equipment is established. Compare the predicted labels of the target domain test set with the real labels, and obtain the accuracy of the model's multi-state classification to measure the performance of the model.

4 EXPERIMENTAL VALIDATION

4.1 Experimental Data Explanation

This article uses PHM2009 Challenge Data from Western Reserve University as experimental data [40] to analyze and validate the proposed EEMD+DCGAN-DANN method, as shown in Figs. 11 and 12. The gearbox test bench includes a 2-horsepower motor (left), a torque sensor/encoder (center), a dynamometer (right), and control electronic equipment, with test bearings used to support the motor shaft. Each pair of meshing gears contains a spur gear and helical gear. The acceleration data are measured near and far from the motor bearing. The Single point of failure is introduced into the test bearing using Electrical discharge machining technology.

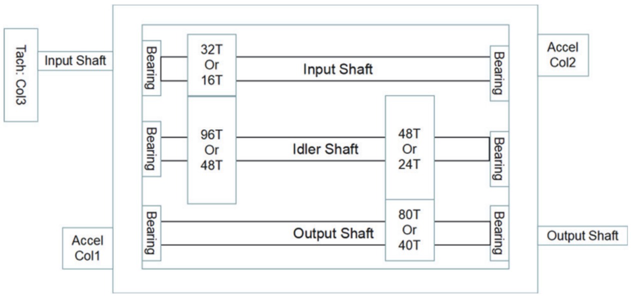


Figure 11 Gearbox structure diagram

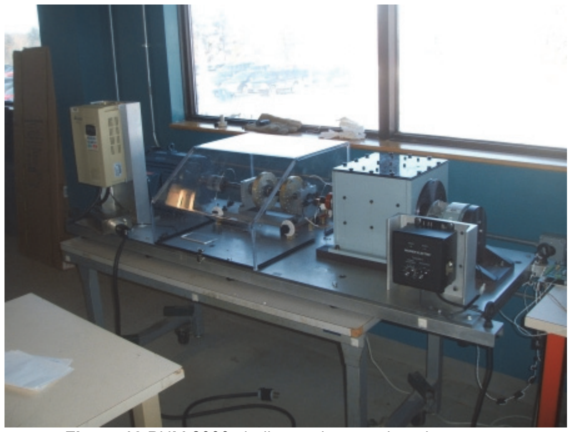


Figure 12 PHM 2009 challenge data test bench structure

The experimental environment is shown in Tab. 2. Faults with a diameter between 0.007 inches and 0.040 inches occurred on the inner raceway, rolling element (i.e. ball), and outer raceway, respectively, and vibration data of the motor load from 0 to 3 horsepower (motor speed 1797 to 1720 rpm) were recorded.

Table 2 Explanation of experimental environment parameters

Experimental environment	Parameter value
Engine speed approval Motor Speed / rpm	1797 / min, 1772 / min, 1750 / min, 1730 / min
Sample Rate	12k (Drive and Fan Bearing Fault Data)
Motor Load (HP)	0, 1, 2, 3
Race Fault Diameter	0.007, 0.014, 0.021
Ball fault diameter	0.007, 0.014, 0.021
Outer ring fault diameter	0.007, 0.014, 0.021

The distribution of different locations collected under different working conditions is shown in Tab. 3.

This chapter sets up different migration tasks to verify the effectiveness of the algorithm in cross domain fault diagnosis problems based on different motor loads and collection locations (where vibration data is collected). It is divided into two experiments as follows:

Experiment 1: Domain adaptation tasks at different collection locations under the same load. Fixed load, all with a load of 01797 revolutions per minute, can be divided into fault training data installed at the driver end and fan end positions according to the different collection locations. In each task, the source domain data and target domain data are tested as spherical fault types.

Experiment 2: Domain adaptation tasks under different loads and the same collection location. The collection position is fixed as the drive end position. According to the different motor loads and speeds, the vibration data under working conditions A, B, C, and D are represented by the load of 0 hp-3 hp. Two domains migrate to each other, setting up three migration tasks: A > B, B > C, and C > D. For example, condition A $\rightarrow$ B indicates that condition A is the source domain (using 0hp data), and condition B is the domain adaptation task of the target domain (using 1hp data).

Table 3 Distribution of different collection locations under different working conditions

Fault status	Normal	Rolling element fault	Load	Collection location
Fault diameter category	0 0	0.007 0.014 0.021 1, 2, 3	——	——
Condition A	600	600 600 600	0 hp	Fan end F0/Drive end D0
Condition B	600	600 600 600	1 hp	Fan end F1/Drive end D1
Condition C	600	600 600 600	2 hp	Fan end F2/Drive end D2
Condition D	600	600 600 600	3 hp	Fan end F3/Drive end D3

4.2 Experiment 1: Same Working Condition Test

Test 1 selects 9 faults of ball, inner race and outer race under the same working condition (load 0, speed 1797 rpm), and the fault migrates from the drive end to the fan end, to verify the accuracy of resisting Transfer learning at different depths under the same working condition. The detailed test description is shown in Tab. 4.

Table 4 Test of drive end and fan end under the same working condition

Test position	Source data	Target Domain Data
	01797 rpm	
Position F0->D0	Fan end, ball failure, 007	Drive end, ball failure, 007
Position F1->D1	Fan end, ball failure, 014	Drive end, ball failure, 014
Position F2->D2	Fan end, ball failure, 021	Drive end, ball failure, 021
Position F3->D3	Fan end, inner ring fault, 007	Drive end, inner ring failure, 007

1) Edge feature extraction

This section uses 12K Driver End at the driver end as the source domain for training on the edge side, and 12K Fan End at the fan end as the target domain for feature extraction of fault types at different locations under four different operating conditions. Due to the different positions of the driver and fan ends, the signal characteristics exhibited by faults are completely different. Therefore, it is considered to migrate the fault learning algorithm of the driver end to the fan side to ensure that the fault diagnosis model can adapt to the driver and fan ends. The load on the drive end and fan end is 0, and the speed is 1797/min. Under the conditions of 0.007, 0.014, and 0.021 fault diameters for the bearings at the fan end and fault end, there are 9 sets of fault data for the inner ring, sphere, and outer ring. Each set of sample signals contains 120000 sampling points (continuously collected for 10 seconds), with an average of 12000 * 60/1797 = 400 data points per cycle, forming 120000 * 10/400 = 3000 samples. The IMF sample components of the vibration acceleration sample signal of a single motor rotating periodic bearing at the source domain drive end after EEMD are shown in Fig. 13.

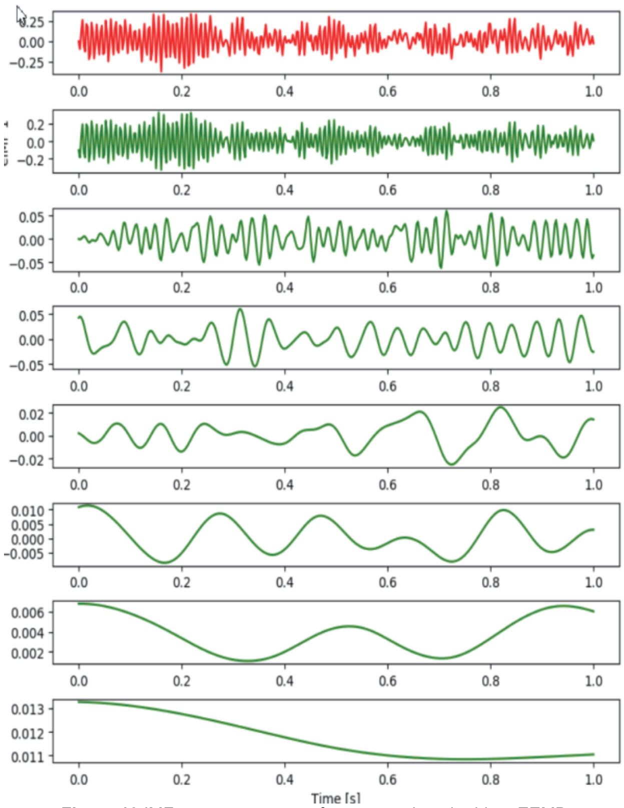


Figure 13 IMF component map after source domain driver EEMD decomposition

Different IMF components contain different time scales, which can display the characteristics of the signal at different resolutions. The fault feature information generated by bearings is mainly contained in the first few IMF components. When local damage occurs to the rolling bearing, an impact component is generated, which usually contains fault feature information. Therefore, kurtosis is selected as the judgment standard for effective IMF, and the kurtosis of each IMF component is shown in Fig. 14.

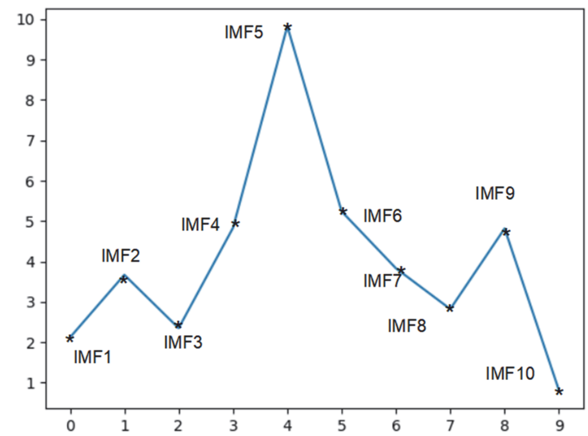


Figure 14 Kurtosis diagram of IMF components at the source domain driver end

Usually, the larger the kurtosis, the more impact components are included in the signal. Based on the kurtosis threshold, the six components with the highest kurtosis are selected as the effective IMF, which contains more vibration feature information generated by bearing faults. Then, stack the 7 IMF components with obvious fault characteristics in descending order of kurtosis to form a multi-channel sample. The sample data dimension changes from 3000 * 400 to 3000 * 7 * 400. Perform the above operations on all sample signals from the source and target domains to create a dataset, and divide the dataset into source domain training and testing sets, target domain training and testing sets. The final dataset obtained is shown in Tab. 5.

Table 5 Source and target domain datasets

Source domain bearing status (drive end)	Fault category	Fault diameter	Training Set	Test Set	Label	Target domain bearing status (fan end)
Driver_ Spur0	Ball failure	007	210	90	0	End_ Spur0
Driver_ Spur1	Inner ring failure	014	210	90	1	End_ Spur1
Driver_ Spur2	Outer ring fault	021	210	90	2	End_ Spur2
Driver_ Spur3	Ball failure	007	210	90	3	End_ Spur3

As shown in the above table, there are three types of faults selected for the source domain dataset and the target domain dataset, including the driver end sphere, inner ring, and outer ring faults.

2) Cloud based adversarial network training

Cloud based adversarial network training utilizes different fault characteristics at the driver and fan ends to conduct DCGAN-DANN network adversarial training in the cloud. After training new network layer parameters, they are validated using source domain and target domain test sets, respectively. In the training process, the number of parameter selection iterations is 200, and the number of gradient descent samples (batch_size) is 50. The model training is divided into three network models, feature extraction network, label Classful network, source domain and target domain Classful network. The corresponding parameters are shown in Tab. 6.

Table 6 Explanation of training parameters for cloud countermeasures network

Network model	Network Description	Input	Output
Feature extraction network	First layer convolution 1. Convolutional f_ Cov1 2. Standardization f_ Bn1 3. Maximum pooling f_ Pool1 4. Activation function f_ Releu1	[50, 1, 7400] Batch_ Size = 50 Input_ Channels = 7 Height = 7 Width = 400	Output_ Channels = 16 Kernel_ Size = [1, 40] BatchNorm2d (16) MaxPoole2d (1, 2) Relu (True)
	Second layer convolution 1. Convolutional f_ Cov2 2. Standardization f_ Bn2 3. Maximum pooling f_ Pool2 4. Activation function f_ Releu2	Input_ Channels = 16	Output_ Channels = 32 Kernel_ Size = [1, 40] BatchNorm2d (32) MaxPoole2d (1, 2) Relu (True)
	Third layer convolution 1. Convolutional f_ Cov3 2. Standardization f_ Bn3 3. Eliminate f_ Drop3 4. Maximum pooling f_ Pool3 5. Activation function f_ Releu3	Input_ Channels = 16	Output_ Channels = 32 Kernel_ Size = [1, 40] BatchNorm2d (64) Dropout2d () MaxPoole2d (1, 2) Relu (True)
Label Classful network	Fully connected layer 1. Fully connected c_ Fc1 2. Standardization c_ Bn1 3. Activation function c_ Releu1 4. Eliminate c_ Drop1	[64 * 1 * 15, 100]	BatchNorm1d (100) Relu (True) Dropout2d ()
	Fully connected layer 1. Fully connected c_ Fc2 2. Standardization c_ Bn2 3. Activation function f_ Releu3	[100, 100]	BatchNorm1d (100) Relu (True)
	Fully connected layer 1. Fully connected c_ Fc3 2. Classification function c_ Soft	[100, 9]	Softmax()
Source domain and target domain Classful network	Fully connected layer 1. Fully connected d_ Fc1 2. Standardization d_ Bn1 3. Activation function d_ Releu1	[64 * 1 * 15, 100]	BatchNorm1d (100) Relu (True)
	Fully connected layer 1. Fully connected d_ Fc2 2. Classification function d_ Soft	[100, 2]	Softmax()

After each round of training, validation is conducted. Finally, the classification accuracy of the model is tested by the test set. After 100 iterations, the accuracy of the source and target domains is shown in Fig.15.

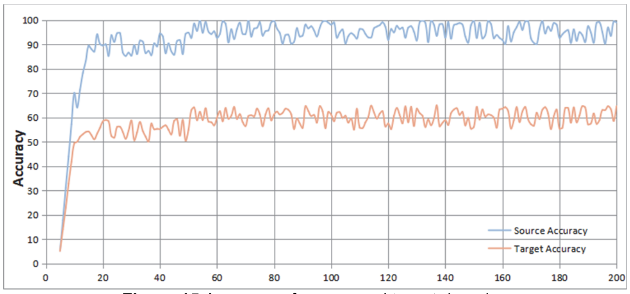


Figure 15 Accuracy of source and target domains

From the above figure, it can be seen that the accuracy of the source domain remains above 90% after the 20th iteration of training, and after 50 iterations, it reaches above 95%. In the end, the accuracy of the model for the source domain test set reaches 95.8%. There is a significant difference in the target domain, with the first 60 iterations fluctuating between 40% and 60%, and decreasing as the accuracy of the source domain improves. After the 60th iteration, it gradually increases and the accuracy is controlled above 60%. Finally, the accuracy of the model on the target domain test set reaches 61.3%.

4.3 Experiment 2: Variable Operating Condition Test

Test 2 is a variable condition experiment to verify the model migration capability of the bearing under different load conditions. In order to show the effect of model

Transfer learning, the model is migrated under load conditions of 0, 1, 2, and 3 respectively. A represents 0 load, 1797 rpm, B represents 1 load, that is, 0.75 kw load, 1772 rpm, C represents 2 load, that is, 1.50 kw load, 1750 rpm, D represents 3 load, 2.20 kw load, 1740 rpm. A → B → CC → D three sets of variable condition migration tests. The detailed test description is shown in Tab. 7.

Table 7 Variable condition test

Test	Source Domain Data	Target Domain Data
A ->B	01797 r/min	0.75 kw, 1772 r/min
B ->C	0.75 kw, 1772 r/min	1.50 kw, 1750 r/min
C ->D	1.50 kw, 1750 r/min	2.20 kw, 1730 r/min

In Experiment 2, the edge features and cloud model were preprocessed on the edge side using the EEMD method under the same working condition. The DCGAN-DANN model was trained on the cloud.

$$G(x) = \text{Desriminator}\left(\text{Generator}(x)\right)$$

The network is trained in batches, and the structure of the DCGAN network is shown in Tab. 8.

Table 8 DCGAN Network Structure Description

Grouping	Network Model	Input	Output
Generator	First layer connection layer 1. Fully connected d_ one 2. Activation function ReLU 2. Standardization bn_ one 3. Transform Shape Reshape	(None, 100)	(None, 64, 256)
	Second layer convolutional layer 1. One dimensional convolution CNN_ one 2. Activation function ReLU 3. Standardization bn_ two	(None, 64, 256)	(None, 128, 128)
	Third layer convolutional layer 1. One dimensional convolution CNN_ two 2. Activation function ReLU 3. Standardization bn_ three	(None, 64, 256)	(None, 128, 128)
	Fourth layer convolutional layer 1. One dimensional convolution CNN_ three 2. Activation function Tanh	(None, 256, 128)	(None, 512, 1)
Discriminator	First layer convolution 1. One dimensional convolution CNN_ four 2. Activation function ReLU 3. Dropout	(None, 512, 128)	(None, 512, 128)
	Second layer convolution 1. One dimensional convolution CNN_ six 2. Activation function ReLU 3. Dropout	(None, 256, 128)	(None, 512, 64)
	Third layer fully connected 1. Flatten 2. Dropout 3. Fully connected Dense 4. Sigmaid	(None, 512, 64)	(None, 32768) (None, 1)

According to the test of condition 1, target 1/100 data points are taken as labeled data, that is, 120000/100 = 1200 samples are taken as labeled data, and the rest are taken as unlabeled test setdata.10 samples are randomly taken as a data batch, and a group of Gaussian noise with the shape of (10100) and subject to Normal distribution is generated in a random space with the step size of 64, and the noise is sampled as signal data with the shape of (10512,1). Label the generated signal as 1, label the real signal as 0, and add random noise to the label. Mix the generated signal with the real signal and substitute it into the training discriminator. Both the model generator and discriminator use Adam with a Learning rate of 0.0001 as the optimizer. By adding a batch normalization layer to the generator, the discriminator continuously uses the Dropout layer to prevent the model from over fitting. The DANN network model is tested under the same working condition.

Taking test condition A (01797 r/min) → B (0.75 kw, 1772 r/min) as an example, as the number of training batches increases, the accuracy of the fault label classifier training set gradually approaches 99.95% and tends to stabilize in 60 batches, and the accuracy of the domain discriminator also gradually approaches 99.98%. Under the action of the reverse gradient layer, the higher the accuracy of the domain discriminator, the more effective the network countermeasures are, prove that the GCGAN-DANN adversarial network successfully projects the source and target domain features into the same feature space, and the accuracy, accuracy, and recall of the three operating conditions are shown in Tab. 9.

Table 9 Accuracy of variant condition training

Test	Accuracy / %	Accuracy / %	Recall rate / %
A ->B	99.65%	99.38%	100%
B ->C	99.51%	99.89%	99.84%
C ->D	99.86%	99.45%	99.98%

The GCGAN-DANN adversarial network can maintain an average accuracy of over 99% for cross domain fault identification under different operating conditions. The network has strong discrimination ability for 7 types of labels. When the number of samples in the source and target domains is unbalanced, the model can maintain a high fault label recognition rate.

5 COMPREHENSIVE RESULTS AND DISCUSSION

In order to evaluate the effectiveness of the DCGAN-DANN algorithm in cross domain fault diagnosis problems, various deep learning methods were used for comparison in this experiment, including:

1) Convolutional neural network (CNN): as the benchmark model, the CNN network is trained on the source domain data, tested in the target domain, and compared with the domain adaptation technology.

2) Deep Domain Confusion (DDC): using the traditional Transfer learning method, add a domain verification layer at the last layer of the network based on the AlexNet network, calculate the minimum difference MMD between the two domains, establish a domain confusion Loss function, and update the network parameters in order to reduce the distribution difference between the source domain and the target domain [34].

3) Domain Adversarial Training of Neural Networks (DANN): DANN is a core algorithm based on adversarial networks to solve domain adaptation problems. The difference from DCGAN-DANN adopted in this article is that it does not generate adversarial training for target domain samples, and directly uses source and target domain sample data to search for similar or identical features, thereby completing the classification task of the target domain.

Further explain the network and training parameter settings of the above algorithm and DCGAN-DANN algorithm:

1) Set epoch to 200 for all algorithms, batch_ Set the size to 50;

2) The structure of the CNN algorithm is designed using the AlexNet network to ensure consistency with the DDC algorithm infrastructure;

3) The network structure and optimization algorithm of DDC algorithm are set according to the network layer structure in the corresponding references;

4) Both DANN algorithm and DCGAN-DANN algorithm are set according to the structural parameter design described in 3.2.2 of this article.

The following is an analysis of two experimental scenarios.

1) Experiment 1: Same working condition experiment

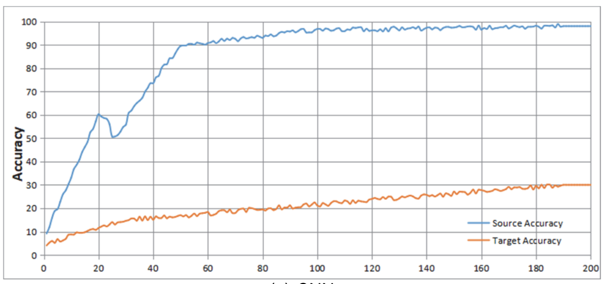
The comparison using different methods such as CNN, DDC, DANN, and DCGAN-DANN is shown in Tab. 10.

Table 10 Accuracy of Different Methods under the Same Working Condition

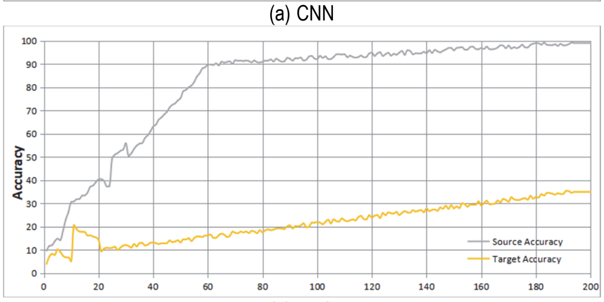
Migration task	Method			
	CNN	DDC	DANN	DCGAN-DANN
Position F0->D0	21.45	28.14	43.45	89.56
Position F1->D1	30.41	32.55	48.93	85.14
Position F2->D2	25.12	24.56	51.42	88.51
Position F3->D3	31.51	35.61	45.15	90.25

From the experimental results, it can be further found that:

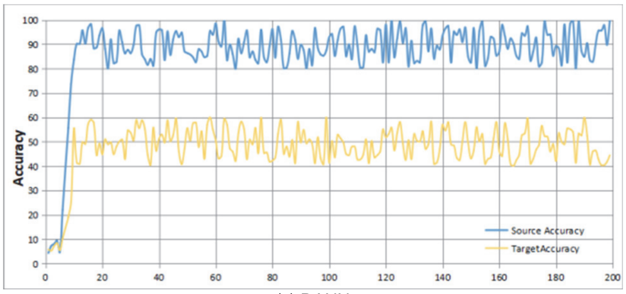
In migration tasks with the same load and different collection locations, the average accuracy of the DDC algorithm is slightly better than that of the CNN algorithm, with an average accuracy of about 3.1% higher. The DANN algorithm is about 17% higher than DDC. The DCGAN-DANN algorithm has the highest accuracy, reaching 88.36%, which is about 41% better than the DANN algorithm. For the fault diagnosis task of this experiment, the overall performance is DCGAN-DANN > DANN > DDC > CNN.



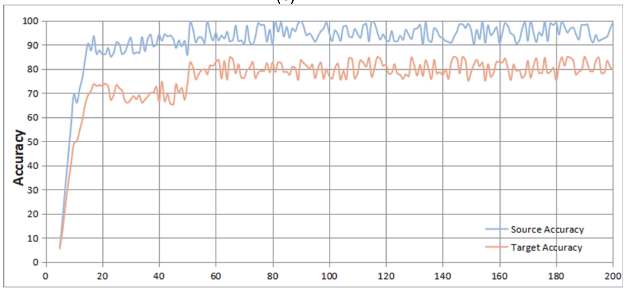
(a) CNN



(b) DDC



(c) DANN



(d) DCGAN-DANN

Figure 16 Validation set accuracy of different algorithms for migration task positions F0->D0

Taking the migration task position F0 → D0 as an example, Fig. 16 shows the accuracy curves of multiple methods iterating with epoch in fault diagnosis at different acquisition positions under the same load. From the perspective of Rate of convergence and convergence stability, CNN and DDC algorithm have the fastest Rate of convergence. From the perspective of convergence accuracy, DCGAN-DANN algorithm has the highest convergence accuracy, and DCGAN-DANN and DANN algorithm have the worst stability. This is because of the confrontation training. Oscillation instability has little impact on the final model results.

2) Experiment 2: Variable operating conditions experiment

The distribution of classification accuracy under different loads and the same collection location under variable operating conditions is shown in Tab. 11.

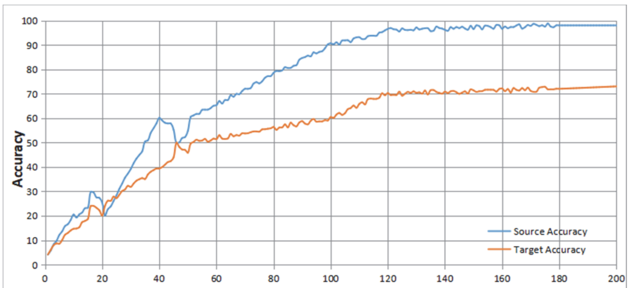
Table 11 Classification accuracy under different loads and the same collection location / %

Migration task	Method			
	CNN	DDC	DANN	DCGAN-DANN
A ->B	70.15	76.81	98.10	99.14
B ->C	67.45	69.41	97.24	99.56
C ->D	69.14	77.01	95.14	99.91

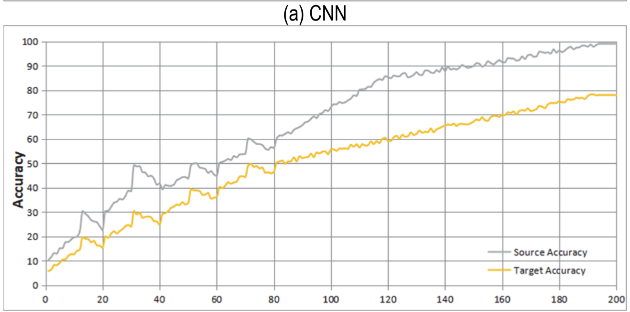
From the experimental results, it can be found that:

1) In the migration tasks with different loads at the same acquisition location, the DANN and DCGAN-DANN using the Transfer learning method of domain adaptation technology are superior to the Convolutional neural network CNN and DDC methods without domain adaptation in fault diagnosis performance. It is proved that Transfer learning can find the best balance point by introducing the domain adaptive loss term and judging the confrontation between the Loss function and the Loss function by labeling the loss function and the domain, so as to realize the fault diagnosis task in the case of a small number of samples in the target domain.

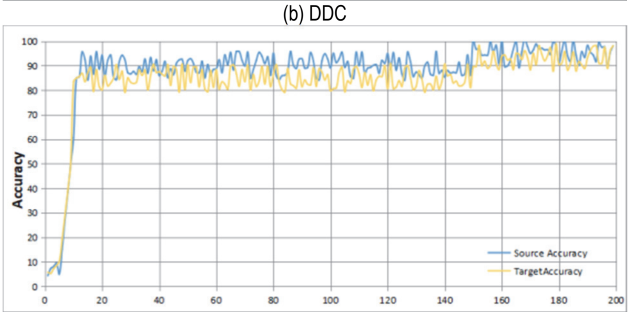
2) Both DANN and DCGAN-DANN algorithm proposed in this paper are deep Transfer learning methods based on Generative adversarial network, and their average accuracy is as follows: DCGAN-DANN>DANN, with an average accuracy of about 2.73% higher. The accuracy curves of different algorithm validation sets for the migration task under condition A > B are shown in Fig. 17.



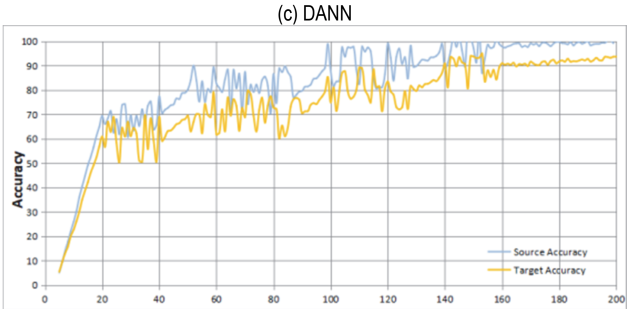
(a) CNN



(b) DDC



(c) DANN



(d) DCGAN-DANN

Figure 17 Accuracy curve of different algorithm validation sets for migration task condition A > B

Taking the migration task scenario A → B as an example, Fig. 4-4 shows the validation set accuracy curves of multiple algorithms iterating with epoch in Experiment 1 fault diagnosis. The following results can be obtained from them:

1) Fig. 17a Convolutional neural network CNN is adopted without introducing domain adaptation technology, and the training accuracy of the source domain is close to 99.8%, but the training accuracy of the target domain is the lowest among the four algorithms, indicating that CNN cannot adapt to the differences between different domains;

2) In Fig. 17b, the DDC algorithm belongs to the MMD metric domain adaptation technology, with good training speed and stability, but low accuracy in the target domain;

3) Figs. 17c and d use the accuracy curves of the DANN algorithm and the DCGAN-DANN algorithm. Compared to the above two algorithms, the training accuracy curves in the source and target domains have a higher degree of overlap and greater volatility. This is because the use of adversarial networks for training in these two algorithms exhibits a high degree of volatility. After the 150th iteration, the accuracy of DCGAN-DANN is higher than DANN, with an average of 5.5% higher than DANN. The results show that the training accuracy of DCGAN-DANN algorithm is better than that of DANN.

6 CONCLUSION

Starting from the background of industrial internet and deep learning based rolling bearing fault diagnosis, this article explores a cloud edge collaboration[37]+adversarial network fault diagnosis method based on industrial internet platform[35][36]. On the one hand, due to the superposition of signals on the industrial site, nonlinear and unstable, it is difficult to obtain fault samples, resulting in difficulty in extracting fault features. On the other hand, the collection of fault samples is limited, and due to changes in equipment operating conditions and different signal collection locations, the data distribution of collected vibration signals may vary, leading to a decrease in diagnostic accuracy and insufficient generalization ability in cloud models. At the same time, the massive data and fault diagnosis algorithms in the cloud require rich computing and storage resources. There is a contradiction between the traditional Convolutional neural network method and the real-time response requirements of fault diagnosis. EEMD and Generative adversarial network, as the research focus of generative models in recent years, can be used for edge side fault feature extraction. Their network countermeasure training mechanism can enhance cloud sample data and can be applied to Transfer learning. This paper innovatively proposes a fault diagnosis method based on cloud edge collaboration + confrontation network Transfer learning for the above two problems, which realizes edge side fault data preprocessing, cloud fault migration online fault training and diagnosis, and improves the accuracy of rate fault model. The main work of this article is as follows:

1) At the edge side, in view of the serious superposition and coupling of vibration signals, and the nonlinear and non-stationary characteristics of vibration signals, EEMD analysis and processing of equipment fault signals are carried out using the edge Real-time computing advantages to achieve line fault diagnosis feature extraction, determine the effective IMF component through kurtosis, and build a data set with more concentrated vibration information, Improved the adversarial training performance and online fault diagnosis accuracy of the cloud based DCGAN-DANN network.

2) In the cloud, aiming at the problem of fault diagnosis in the case of small sample size, unbalanced data, and cross working conditions and cross acquisition locations, a new DCGAN-DANN Transfer learning model

is proposed by combining the Generative adversarial network with the domain adaptive problem. By establishing the network structure of feature extractor, domain discriminator, and tag predictor, DCGAN-DANN forms the confrontation training of feature extractor and domain discriminator, the training mode of label predictor assisted classification. Through the PHM2009 test data test of Western Reserve University, the proposed cloud edge collaboration + DCGAN-DANN fault diagnosis method is superior to the traditional fault diagnosis methods CNN and DDC under the two migration experiments of cross working conditions and cross acquisition locations. The EEMD + DCGAN-DANN based on cloud edge collaboration + Transfer learning has higher accuracy and better stability of the diagnosis results.

7 REFERENCES

[1] Strachey, C. (1959). Time sharing in large fast computers. *International Conference on Information Processing.*

[2] Xue, M. A., Chenglin, W., Xue, M. A., & Chenglin, W. (2021). An asynchronous quasi-cloud/edge/client collaborative federated learning mechanism for fault diagnosis. *Chinese Journal of Electronics.* https://doi.org/10.1049/cje.2021.07.018

[3] Sun, M. & Bao, T. (2023). An Efficient Task Offloading Strategy based on Deep Reinforcement Learning in Edge Cloud Collaborative Computing. *Chinese Journal of Electronics.*

[4] Yoshinski, J., Clune, J., Bengio, Y. et al. (2014). How transferable are features in deep neural networks. *Advances in neural information processing systems*, *27*, 3320-3328.

[5] Weiss, K., Khoshgoftaar, T. M., & Wang, D. D. (2016). A survey of transfer learning. *Journal of Big Data*, *3*(1), 1-40.

[6] Pan, S. J. & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345-1359.

[7] Ren, Z. & Lee, Y. J. (2018). Cross domain self supervised multi task feature learning using synthetic images. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[8] Deng, W., Zheng, L., Kang, G. et al. (2018). Image image domain adaptation with preserved self-similarity and domain similarity for person identification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, *1*(2), 6.

[9] Hong, W., Wang, Z., Yang, M. et al. (2018). Conditional Generative Adversarial Network for Structured Domain Adaptation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1335-1344.

[10] Socher, R., Ganjoo, M., Manning, C. D. et al. (2013). Zero shot learning through cross modal transfer. *Advances in neural information processing systems*, 935-943.

[11] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., & Laviolette, F. et al. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, *17*(1), 2096-2030. https://doi.org/10.1007/978-3-319-58347-1_ ten

[12] Yu, C., Wang, J., Chen, Y. et al. (2019). Transfer Learning with Dynamic Adversarial Adaptation Network. *Computer Science*, 1-9.

[13] Tzeng, E., Hoffman, J., Saenko, K. et al. (2017). Adversarial Discriminative Domain Adaptation. *IEEE Transactions on Image Processing*, *29*, 2622-2637.

[14] Legan, Y. & Bengio, Y. (1995). Convolutional Networks for Images, Speech, and Time Series. *Handbook of Brain Theory & Neural Networks*, 1-14.

[15] Zhang, Y. & Gong, Z. (2022). Joint study on health status assessment and degradation trend prediction of industrial equipment. *Chinese Science: Technical Science*, *52*(01), 180-197.

[16] Jiaxian, C. & Wentao, M. (2021). Method for predicting residual life of bearings based on deep temporal feature transfer. *Control and Decision*, *36*(07), 1699-1706.

[17] Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., & Zheng, Q. et al. (1998). The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings Mathematical Physical & Engineering Sciences*, *454*(1971), 903-995. https://doi.org/10.1098/rspa.1998.0193

[18] He, Q., Peng, L., & Kong, F. (2012). Rolling bearing localized defect evaluation by multiscale signature via empirical mode decomposition. *Journal of Vibration & Acoustics*, *134*(6), 061013. https://doi.org/10.1115/1.4006754

[19] Lei, Y., He, Z., & Zi, Y. (2009). Application of the EEMD method to rotor fault diagnosis of rotating machinery. *Mechanical Systems & Signal Processing*, *23*(4), 1327-1338.

[20] Day, O. & Khoshgoftaar, T. M. (2017). A survey on heterogeneous transfer learning. *Journal of Big Data*, *4*(1), 29. https://doi.org/10.1186/s40537-017-0089-0

[21] Xinpeng, L., Xidao, L., Yuxiang, X., & Mingzhe, H. (2018). Overview of transfer learning research and algorithm. *Journal of Changsha University*, *32*(5), 33-36

[22] Fei, S., Chao, C., Yan, R., & Gao, R. X. (2015). Bearing fault diagnosis based on SVD feature extraction and transfer learning classification. *Prognostics & System Health Management Conference (PHM)*, 1-6.

[23] Xie, J., Zhang, L., Duan, L., & Wang, J. (2016). On cross-domain feature fusion in gearbox fault diagnosis under various operating conditions based on Transfer Component Analysis. *IEEE International Conference on Prognostics and Health Management (ICPHM).* https://doi.org/10.1109/ICPHM.2016.7542845

[24] Lu, W., Liang, B., Yu, C., Meng, D., & Tao, Z. (2016). Deep model based domain adaptation for fault diagnosis. *IEEE Transactions on Industrial Electronics*, *PP*(99), 1-1. https://doi.org/10.1109/TIE.2016.2627020

[25] Han, T., Liu, C., Yang, W., & Jiang, D. (2019). A novel adversarial learning framework in deep convolutional neural network for intelligent diagnosis of mechanical faults. *Knowledge-Based Systems*, *165*(FEB.1), 474-487. https://doi.org/10.1016/j.knosys.2018.12.019

[26] Goodfellow, I., Pouget-Abadie, J., Mirza, M. et al. (2014). Generative Adversarial Networks. *Statistics.* https://doi.org/https://arxiv.org/abs/1406.2661

[27] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *Computerence.* https://doi.org/10.48550/arXiv.1511.06434

[28] Mirza, M. & Osindero, S. (2014). Conditional generative adversarial nets. *Computer Science,* 2672-2680. https://doi.org/10.48550/arXiv.1411.1784

[29] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *IEEE.* https://doi.org/10.1109/ICCV.2017.244

[30] Ganin, Y. & Lempitsky, V. (2014). Unsupervised domain adaptation by backpropagation. *JMLR.org.* https://doi.org/10.48550/arXiv.1409.7495

[31] Ruohui, H., Min, Z., & Wenxin, X. (2022). Cross domain fault diagnosis of rolling bearings based on DCGAN and DANN networks. *Vibration and Shock*, *41*(06), 21-29.

[32] Zhang, Y., Ren, Z., & Zhou, S. (2020). An intelligent fault diagnosis for rolling bearing based on adversarial semi-supervised method. *IEEE Access*, *PP*(99), 1-1. https://doi.org/10.1109/ACCESS.2020.3016314

[33] Jiao, J., Zhao, M., & Lin, J. (2019). Unsupervised adversarial adaptation network for intelligent fault diagnosis. *IEEE Transactions on Industrial Electronics*, *PP*(99), 1-1. https://doi.org/10.1109/TIE.2019.2956366

[34] Tzeng, E., Hoffman, J., Zhang, N. et al. (2014). Deep Domain Fusion: Maximizing for Domain Invariance. *Computer Science*, 1-9.

[35] Simon, J., Trojanová, M., Hošovský, A. & Sárosi, J. (2021). Neural Network Driven Automated Guided Vehicle Platform Development for Industry 4.0 Environment. *Tehnički vjesnik*, *28*(6), 1936-1942. https://doi.org/10.17559/TV-20200727095821

[36] Marinas, M., Dinu, M., Socol, A., & Socol, C. (2021). The Technological Transition of European Manufacturing Companies to Industry 4.0. *Economic Computation and Economic Cybernetics Studies And Research*, *55*(2), 23-42. https://doi.org/ 10.24818/18423264/55.2.21.02

[37] Shi, W., Tang, D., & Zou, P. (2021). Research on Cloud Enterprise Resource Integration and Scheduling Technology Based on Mixed Set Programming. *Tehnički vjesnik*, *28*(6), 2027-2035. https://doi.org/10.17559/TV-20210718091658

[38] Nath, B., Kumbhar, C., & Khoa, B. T. (2022). A study on approaches to neural machine translation. *Journal of Logistics, Informatics and Service Science, 9*(3), 271-283. https://doi.org/10.33168/LISS.2022.0319

[39] Madhavi, S. & Hong, S. P. (2022). Anomaly detection using deep neural network quantum encoder. *Journal of Logistics, Informatics and Service Science*, *9*(3), 118-130. https://doi.org/10.33168/LISS.2022.0207

[40] Ciurea, C., Chiriţă, N., & Nica, I. (2022). A Practical Approach to Development and Validation of Credit Risk Models Based on Data Analysis. *Economic Computation and Economic Cybernetics Studies and Research*, *56*(3), 51-68. https://doi.org/ 10.24818/18423264/56.3.22.04

Contact information:

Ping ZOU, Senior Engineer
(Corresponding author)
1. School of Economics and Management,
Beijing Jiaotong University, Beijing 100040, China
E-mail: 18113038@bjtu.edu.cn
2. Sany Group Co.,Ltd.,
Changsha, Changsha County, Hunan Province, 430100, China
E-mail: ziapple@126.com

Zhenji ZHANG, Professor, Doctoral Supervisor
School of Economics and Management,
Beijing Jiaotong University, Beijing 100040, China
E-mail: zhangzhenji@bjtu.edu.cn

Lei JIANG, Engineer
China Railway Information Technology Group Co., Ltd.,
Building 1, Yard 5, Beifengwo Road,
Haidian District, Beijing, 100844, China
E-mail: ginajiang1984@163.com