Weighted joint matching adaptive network for fault diagnosis of wind turbines in the new wind farm (PHM2021)

Guangyao Zhang

Shanghai Jiao Tong University <https://orcid.org/0000-0002-2739-4201>

Yanting Li (✉ ytli@sjtu.edu.cn)

Shanghai Jiao Tong University

Research Article

Keywords: wind turbine, fault diagnosis, transfer learning, domain adaptation, SCADA, limited data

Posted Date: March 7th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1312264/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Weighted joint matching adaptive network for fault diagnosis of wind turbines in the new wind farm (PHM2021)

Guangyao Zhang¹

School of Mechanical
Engineering

Shanghai Jiao Tong University
Shanghai, China
zhanggy23@sjtu.edu.cn

Yanting Li^(2,*)

School of Mechanical
Engineering

Shanghai Jiao Tong University
Shanghai, China
ytli@sjtu.edu.cn

Abstract—In recent years, intelligent fault diagnosis methods based on deep learning have been widely used in the wind power industry. These methods mostly rely on sufficient labeled data to support model training. However, new wind farms often face the challenge of insufficient labeled data, which will hinder the practical application of the deep learning methods. In addition, data distributions of different wind farms are usually different, so the model trained by other wind farm data cannot be directly used for fault diagnosis of new wind farms. In view of the lack of labeled data in new wind farms, a transfer learning method: weighted joint matching adaptive network (wJMAN) is proposed. This method realizes the alignment of the source domain (other existing wind farms) and the target domain (the new wind farm) through sub-domain matching and joint distribution adaptation. Therefore, the information of other wind farms can be more effectively used in the construction of the fault diagnosis model of the new wind farm. In addition, in view of the imbalance between classification loss and adaptive loss in existing domain adaptation methods, an adaptive loss weight method is proposed, which makes the classification loss and adaptive loss more balanced and converges faster in the optimization process. The proposed wJMAN method is tested with the operating data from 4 real wind farms. The results show that wJMAN has higher diagnostic accuracy than common transfer learning methods.

Keywords—wind turbine, fault diagnosis, transfer learning, domain adaptation, SCADA, limited data

I. INTRODUCTION

The massive consumption of fossil energy has led to the increasing impact of the climate crisis, which has caused a large number of countries and regions to seriously implement the net-zero plan. The wind energy, as a clean power, has developed rapidly. According to the Global Wind Energy Report 2021 issued by the Global Wind Energy Council, 2020 has witnessed global wind power installations exceeding 90GW, an increase of 53% compared to 2019 [1].

With the rapid development of the wind power industry, more and more new wind farms are being built, and the number of wind turbines as well as the core equipment for wind power has also increased. Most wind turbines are installed on the mountains, seaside and other harsh natural environments. The

rapid changes in climate bring unstable loads to the wind turbines, all of which lead to frequent failures of the wind turbines[2]. The ever-increasing number of failure-prone wind turbines brings more and more maintenance economic costs and time loss to wind farm owners. Fault diagnosis of wind turbines based on operation data can realize remote fault location and type identification, and therefore has great practical value.

In recent years, with the rapid development of artificial intelligence, intelligent fault diagnosis methods that introduce artificial intelligence algorithms have become one of the most popular fault diagnosis directions. Among them, fault diagnosis methods based on deep learning have been verified to have excellent performance. Common deep learning models, such as convolutional neural networks (CNN) [3, 4], deep autoencoders (DAE) [5, 6], recurrent neural networks (RNN) [7, 8], long short-term memory (LSTM) [9, 10], deep belief networks (DBN) [11] etc., have been applied to wind turbine fault diagnosis research because they usually have a strong ability to automatically learn hidden fault features. However, deep learning methods have heavy data dependence, which means that sufficient labeled data is required to support model training. Otherwise, the model either cannot converge or has a poor detection performance. Compared to wind farms with considerably long running time, new wind farm has limited normal data and even little fault data. New turbines are not prone to failure, which will result in little fault data and incomplete fault types. Therefore, the challenges that the most new wind farm must face include insufficient wind turbine operation data, fault data and incomplete fault types, which bring difficulties to the deep learning-based fault diagnosis of wind turbines.

The essence of solving the challenges of fault diagnosis of wind turbines in new wind farms is to solve the problem of small samples. The methods to solve the small sample problem can be roughly divided into two categories: one is to solve from the inside, including data expansion and utilization. However, the ability of data expansion to solve the problem of small samples is limited. In presence of little data, it is usually impossible to rely solely on data expansion to improve the detection ability of the model. The other is to solve from the outside. Transferring external information to the target domain

(the new wind farm) through transfer learning methods is a common option [12].

The fine-tuning of the existing model is one of the simplest transfer learning methods. Its working principle is to perform follow-up training on the basis of the trained model. Yang et al. [13] used sufficient labeled old gas turbine data to train CNN, and then used the new gas turbine data to fine-tune the trained CNN model. After experimental verification, the final fine-tuned CNN model has higher diagnostic accuracy than the CNN model trained only on new gas engine data. Li et al. [14] proposed a wind turbine fault diagnosis method based on fine-tuning convolutional autoencoders, and the experiments of real wind turbine data proved that the method can transfer the old wind turbine data to the target new wind turbine fault diagnosis, and improve the diagnostic accuracy. However, the fine-tuning method has limited performance when the distribution of source domain and target domain are too different. For a new wind farm, this kind of fine-tuning method is no longer applicable when the difference between the wind turbine data of the existing wind farm and the target new wind farm data is too large.

The domain adaptive method is a transfer learning method that can better solve the problem of small samples. It constructs invariant feature spaces of different domains, so that the data of these domains with different distributions in the regular space can have the same distribution in the feature space. Domain adaptive methods are also widely used in fault diagnosis. Liao et al. [15] proposed a dynamic distribution adaptation based transfer network (DDATN) for cross domain bearing fault diagnosis. DDATN uses the proposed instance-weighted dynamic maximum mean discrepancy (IDMMD) to represent the distance between target domain and source domain, and numerical experiments show that the proposed DDATN has excellent cross-domain fault diagnosis performance. Wen et al. [16] proposed a deep transfer learning (DTL) method based on domain adaptation and sparse autoencoder. Experiments show that compared to other common machine learning and deep learning methods, this method has better diagnostic accuracy on the Case Western Reserve University dataset. Li et al. [17] combined ensemble learning and domain adaptive methods to propose a new method with higher diagnostic accuracy for unlabeled data of rotating bearings. Although these methods have good performance in fault diagnosis, they still have shortcomings: firstly, these methods are often built for fault diagnosis of high-frequency vibration signals, and cannot handle medium and low frequency data or have poor fault diagnosis for medium and low frequency data; Secondly, in the face of extremely small amounts of data, the accuracy of fault diagnosis needs to be improved.

Although many transfer learning methods have been used to diagnose faults in presence of insufficient labeled data, the diagnostic performance of these methods needs to be improved. Zhang et al. [18] proposed a transfer learning method for new wind farms: joint matching adaptive network (JMAN). Experiments on wind turbine operating data in real wind farms confirmed that JMAN has better diagnostic performance. However, the optimization objective of JMAN is only the simple

addition of the classification loss and the adaptive loss, and does not consider the priorities of the two types of losses in the optimization process. Therefore, this paper proposed a transfer learning method based on JMAN: weighted joint matching adaptive network (wJMAN). The main contributions and innovations of this method are as follows:

- a) By transferring data from other existing wind farms to the target new wind farm, wJMAN can solve the problem of insufficient wind turbine data for the new wind farm to a certain extent.
- b) A sub-domain matching module. Common methods rely on minimizing the distribution distance to achieve the alignment of the source domain and the target domain. This paper introduces a sub-domain matching algorithm that can improve the diagnostic capabilities.
- c) Considering the different priorities of classification loss and adaptive loss in the optimization process, in order to make the two losses more balanced and converge faster, the optimization objective of wJMAN is the adaptive weighted sum of the two types of losses. And this paper presents an adaptive learning strategy for weights.

The rest of the paper is organized as follows. The related works on domain adaptation and deep domain adaptive method are introduced in Section II. In Section III, the overall structural framework and key modules of the proposed wJMAN are introduced in detail. In Section IV, the proposed method is validated by the wind turbine data from 4 real wind farm, and the performance of the method is displayed and analyzed. Finally, the conclusion of this paper is given in Section V.

II. RELATED WORKS

A. Domain Adaptation

As a typical transfer learning method, the domain adaptation transfers the information from the source domain to the target domain by constructing an invariant feature space. The invariant feature space can usually be realized by adaptively minimizing the distance of the feature distribution of the two domains. Maximum mean discrepancy (MMD) [19] is one of the most common standard metric used to measure the distribution distance between different domains. According to the different distribution distances of the source domain and the target domain, domain adaptation methods generally include three types of distribution adaptation: marginal distribution adaptation (MDA) [20], conditional distribution adaptation (CDA) [21] and joint distribution adaptation (JDA) [22]. And these three distribution adaptations are respectively aimed at the alignment of the marginal distribution, conditional distribution and joint distribution of the source and target domains, where joint distribution means that both marginal distribution and conditional distribution are considered at the same time. Specifically, MDA aligns the source and target domains by minimizing the overall MMD, and CDA aligns the source and target domains by minimizing the MMD of each category of the two domains, which is also called conditional MMD (CMMD). JDA aligns the source and target domains by minimizing joint

MMD (JMMD), where JMMD is usually the sum of the overall MMD and CMMD.

Fig. 1. Marginal domain adaptation by MMD

Fig. 2. Conditional domain adaptation by CMMD

Fig. 3. Joint domain adaptation by JMMD

Because MDA only align the overall distribution of the two domains, it is prone to negative transfer when it solves the multi-classification problem. As shown in Fig. 1, although the overall distribution is aligned, the specific categories are not necessarily all aligned. In the process of CDA minimizing CMMD, sometimes different types of MMD conflict with each other, which makes it difficult to converge. As shown in Fig. 2, the optimization process in which the minimum CMMD is the optimization objective is often not able to achieve the minimum MMD of all categories at the same time, or the MMD of all categories can be minimized at the same time but the difference between different categories is too small, which increases the

difficulty of classification. As shown in Fig. 3, in the process of minimizing JMMD by JDA, although the overall and specific category alignment is considered, it may still face similar situations with CDA: one is that the alignment of all categories cannot be achieved at the same time, which is caused by the conflict between categories; the other is that the alignment of the source domain and the target domain affects the classification ability.

B. Deep Domain Adaptive Method

As deep learning methods have been widely verified to have excellent automatic learning ability to obtain features, more and more scholars have introduced deep learning methods into domain adaptation, and proposed a series of deep domain adaptive (DDA) methods. In 2014, Ghifary et al. [23] combined domain adaptation and deep learning models for the first time, and proposed a domain-adaptive neural network (DaNN), which improves the classification accuracy while aligning the source and target domains. However, DaNN relies on a shallow neural network (NN) to extract features, and the transfer performance on complex data needs to be improved. Some scholars further proposed many more complex DDA methods. For example, Tzeng et al. [24] proposed a typical DDA method called deep domain confusion (DDC). Based on trained AlexNet [25] model, DDC inserts an MMD-based adaptive layer in the following fully connected layer, which has a good effect on domain adaptation of the ImageNet dataset. Both DaNN and DDC are deep domain adaptive methods based on MDA. Therefore, both methods may face the phenomenon that the specific categories cannot be fully aligned (Fig. 1).

Long et al. [26] introduced the idea of JDA into deep domain adaptation, and proposed a method called joint adaptive network (JAN), which can reduce the problem of negative transfer to a certain extent and has a better classification effect. Since JMMD can more effectively represent the distribution distance between different domains, by minimizing JMMD to achieve the alignment of source and target domains, JAN has better diagnostic performance than DaNN and DDC, but its diagnostic performance still has a lot of room for improvement. Due to the same challenge faced by JDA (Fig. 3), the diagnostic performance of JAN needs to be improved.

Fig. 4. Domain adaptation by Sub-domain Matching and JMMD

Fig. 5. The proposed joint matching adaptive network overall structure

The wJMAN proposed in this paper is based on JAN and innovatively proposes a sub-domain module. As shown in Fig. 4, the domain adaptation process of wJMAN is as follows: first, the subdomains of the same category in the source domain and the target domain are matched, and then the JMMD of the two domains is minimized, and finally the invariant feature space is obtained. Due to the sub-domain matching, the conflict between the JMMD categories in Fig. 3 is reduced, and the model diagnosis performance is improved.

III. METHODOLOGY

In order to solve the problem of the small amount of wind turbine data in the new wind farm, we were inspired by JAN and proposed the wJMAN model. The wJMAN method includes three main modules: data sub-domain matching, CNN-based feature extraction and JMMD-based optimization. The framework of wJMAN is shown in the Fig. 5. The following describes the data sub-domain matching, CNN-based feature extraction, the JMMD-based optimization objective in detail.

A. Data Sub-domain Matching

The object of the method proposed in this paper is the new wind farm. Therefore, the new wind farm dataset is used as the target domain, and the old wind farm dataset is used as the source domain. Let $D_S = \{x_i^S, y_i^S\}_{i=1}^{n_S} = \{X_S, Y_S\}$, $i = 1, 2, \dots, n_S$, represent the dataset of the source domain, and $D_T = \{x_j^T, y_j^T\}_{j=1}^{n_T} = \{X_T, Y_T\}$, $j = 1, 2, \dots, n_T$, denotes the dataset of the target domain, where x_i^S and x_j^T are the data in form of one-dimensional vectors respectively, y_i^S and y_j^T are respectively the labels of x_i^S and x_j^T .

The data from the source domain and the target domain are matched according to the same label to obtain a matching dataset D_M , and then the matched data are input into the model. The matching dataset D_M and its element acquisition rules are represented as follows:

$$D_M = \{d_i^M\}_{i=1}^{n_M} = \{x_i^{MS}, x_i^{MT}, y_i^M\}_{i=1}^{n_M} = \{X_{MS}, X_{MT}, Y_M\} \quad (1)$$

where n_M is the data size of the matching dataset D_M , and d_i^M is the cell element in D_M . $\forall a \in \{1, 2, \dots, n_S\}$, $\forall b \in \{1, 2, \dots, n_T\}$, if $y_a^S = y_b^T$, then the detail elements in the matching dataset D_M are shown as follows:

$$\begin{cases} x_i^{MS} = x_a^S \\ x_i^{MT} = x_b^T \\ y_i^M = y_a^S \end{cases} \quad (2)$$

where $y_a^S, x_a^S \in D_S = \{X_S, Y_S\}$ and $y_b^T, x_b^T \in D_T = \{X_T, Y_T\}$. It should be emphasized that the data of the source domain and target domain which have participated in the matching must not be repeatedly participated in the matching process. That is, $\forall h, k \in \{1, 2, \dots, n_M\}$, if $h \neq k$, then $d_h^M \neq d_k^M$, where $d_h^M = (x_h^{MS}, x_h^{MT}, y_h^M)$ and $d_k^M = (x_k^{MS}, x_k^{MT}, y_k^M)$.

B. Feature Extraction based on CNN

CNN is a deep neural network. Its unique weight distribution structure reduces the complexity of the neural network as well as fulfill feature extraction. The most important convolutional layer in the CNN structure consists of a convolution kernel. The convolution process is shown in Equation (3):

$$x_j^l = f\left(\sum_{i \in M_j^{l-1}} x_i^{l-1} * w_{ij}^l + b_j^l\right) \quad (3)$$

where $*$ represents the convolution operator, x_j^l represents the j feature map of the l layer, x_i^{l-1} represents the i feature map of the $(l-1)$ layer, and M_j^{l-1} represents the number set of the feature map related to the j feature map of the l layer in the $(l-1)$ layer. w_{ij}^l represents the element of the convolution kernel that connects the i -th feature map in the $(l-1)$ layer with the j -th feature map of the l layer. b_j^l represents the offset value of the j feature map of the l layer. f is the activation function. and the common ReLU function is used here.

The purpose of adding a pooling layer between the convolutional layers is to reduce computational complexity and enhance the robustness of feature extraction. The pooling process is generally divided into maximum pooling and average pooling, which take the maximum or average value of the receptive field respectively. Taking maximum pooling as an example, assuming that the $(l+1)$ layer is the pooling layer, the expression of the j -th feature map of the $(l+1)$ layer is shown in Equation (4):

$$x_j^{l+1} = \max(\{x_i^l\}_{i=1}^{N_j^l}) \quad (4)$$

where $\max()$ represents the maximum value in parentheses, x_i^l is the i -th feature map of the l layer, and N_j^l the number set of the feature map related to the j feature map of the $(l+1)$ layer in the l layer.

C. Optimization Objective

1) Joint Maximum Mean Discrepancy

The proposed wJMAN method uses JMMD to measure the distribution distance between source domain and target domain. JMMD includes the marginal distribution distance and conditional distribution distance, which are defined as follows:

$$\begin{aligned} J(D_M) &= l_m(D_M) + l_c(D_M) \\ &= \|p(X_{MS}) - q(X_{MT})\| + \|p(X_{MS}|Y_M) - q(X_{MT}|Y_M)\| \\ &= MMD^2[X_{MS}, X_{MT}] + \sum_{c=1}^C MMD^2[X_{MS}^{(c)}, X_{MT}^{(c)}] \end{aligned} \quad (5)$$

where source data X_{MS} and target data X_{MT} follow distribution p and q respectively. It is noted that $p(X_{MS}) \neq q(X_{MT})$, which means source domain and target domain have different marginal distributions. $\|p(X_{MS}) - q(X_{MT})\|$ is the marginal probability distribution distance between source domain and target domain, and $\|p(X_{MS}|Y_M) - q(X_{MT}|Y_M)\|$ is the distance of conditional probability distribution between source domain and target domain. For the convenience of subsequent calculations, the square of the MMD is often used as the distance between the two distribution spaces. C indicates the number of states, $X_{MS}^{(c)}$ and $X_{MT}^{(c)}$ respectively represent the data labeled c of two domains.

JMMD is a distribution metric based on MMD. The premise of solving JMMD is to solve MMD. The key to solving MMD is to find a mapping function ϕ that maximizes the difference between the mean values of the mappings in the two domains of the mapping space. Based on Borgwardt's interpretation of MMD [19] and the previous data matching, the definition of MMD in this paper is given as follows:

$$MMD[X_{MS}, X_{MT}] = \left\| \frac{1}{n_M} \sum_{i=1}^{n_M} \phi(x_i^{MS}) - \frac{1}{n_M} \sum_{j=1}^{n_M} \phi(x_j^{MT}) \right\|_H \quad (6)$$

where ϕ is a nonlinear mapping function in a Reproducing Kernel Hilbert Space H (RKHS). When the value of MMD approaches zero, it means that the distributions of the two domains are the same, that is, the features of the two domains are in the invariant feature space.

2) Classification Learning

The ultimate goal of the model is to achieve a better classification ability in target domain. Therefore, minimizing the distribution distance between the two domains is not enough. It is also necessary to train the model's ability to classify the source domain. In order to realize the classification ability of the proposed JMAN, it is expected that the classification error between the predicted label of the source domain data after the model processing and the true label of the source domain data is minimized. That is:

$$\begin{aligned} l_s(X_{MS}, Y_M) &= \\ &= -\frac{1}{n_M} \sum_{i=1}^{n_M} \sum_{j=1}^{n_l} I\{y_i = j\} \log(P(f(x_i) = j)) \end{aligned} \quad (7)$$

where $f(x_i)$ represents the predicted value of the i -th sample x_i after being processed by wJMAN. P represents the probability that the predicted value $f(x_i)$ is equal to j . When the true label of the i -th sample is equal to j , $I\{y_i = j\}$ is 1, otherwise it is 0. n_l represents the number of categories of training data labels, and n_M represents the number of training data.

3) Overall Optimization Objective

Most deep domain adaptation methods usually just add the two as the overall loss function, as shown below:

$$L = l_s(X_{MS}, Y_M) + J(D_M) \quad (8)$$

where L is the total loss function of the network, $l_s(X_{MS}, Y_M)$ is the conventional classification loss on the source domain data, and here the classification loss chooses cross-entropy loss. The $J(D_M)$ (JMMD) is the adaptive loss of the network.

The importance of classification loss and domain adaptive loss should be different under different values. Specifically, a larger loss means that the corresponding effect of the loss is poor, and the next iterative optimization needs to be focused, so the importance (weight) of the loss is larger. Therefore, we construct a new overall loss function with adaptive weights:

$$L = w_s l_s(X_{MS}, Y_M) + w_J J(D_M) \quad (9)$$

where w_s and w_f represent the weight of supervised learning loss and the weight of domain adaptation, respectively.

In order to conveniently represent the adaptive learning process of these two weights, the overall loss function of the i -th time is expressed as:

$$L^i = w_1^i l_1^i + w_2^i l_2^i = \sum_{m=1}^2 w_m^i l_m^i \quad (10)$$

where L^i represents the overall loss function at the i -th iteration, l_1^i , w_1^i respectively represent the classification loss and its weight at the i -th iteration, and l_2^i , w_2^i represent the domain adaptive loss and its weight at the i -th iteration, respectively.

4) Weight Learning Strategy

In order to express the learning strategy of weights more clearly, a variable τ_m^i is introduced to represent the proportion of the m -th loss of the i -th iteration in the overall loss of the i -th iteration:

$$\tau_m^i = \frac{l_m^i}{\sum_{m=1}^2 l_m^i} \quad (11)$$

If the m -th loss l_m^i accounts for a large proportion at the i -th iteration, that is, τ_m^i is large, then the m -th loss at the next iterative optimization, l_m^{i+1} needs to be considered more seriously, which also means that the weight of the next iterative optimization of l_m^{i+1} , w_m^{i+1} should be larger. Based on the above point of view, this paper gives the detailed learning strategy of the weights:

$$w_m^{i+1} = \begin{cases} \frac{1}{2} [(2\tau_m^i - 1)^3 + 1], & i \geq 1 \\ \frac{1}{2}, & i = 0 \end{cases} \quad (12)$$

where w_m^{i+1} is the weight of the m -th loss at the $(i+1)$ -th iteration. According to the weight learning strategy, when i is equal to 0, the weight w_m^1 of l_m^1 at the first iteration is 0.5, which means that the two losses have the same importance. According to Equation (11) and Equation (12), we easily find that the weight of the classification loss plus the weight of the JMMD loss is equal to 1.

Fig. 6. The relationship between the i -th τ and the $(i+1)$ -th weight of the loss in the overall loss.

Fig. 6 shows the relationship between the i -th τ and the weight of the $(i+1)$ -th loss in the overall loss. Obviously, the weight increases with the increase of τ . In the middle range of τ , the weight change is gentle, while the weight change is steep at both ends of τ . Since τ reflects the proportion of this type of loss in the overall loss, when τ is around 0.5, it means that there is little difference between the two types of losses, and the weights should not be significantly different. When τ is larger or smaller, the larger loss needs to be more weighted, and the smaller loss needs to be smaller, so that the two losses can reach equilibrium and converge faster. So this weight learning strategy adopts the above nonlinear weighting strategy. According to Equation (12), when the value of τ is between 0.3-0.7, the weight range is only 0.496-0.504.

IV. CASE STUDY

A. Datasets

1) Data source

Supervisory control and data acquisition (SCADA) system is used for wind farm data collection, monitoring and generating wind farm performance reports. The wind turbine operating data collected by the SCADA system, also called SCADA data, record a large amount of wind turbine operating variables, such as output power and bearing temperature etc., and some environmental variables, such as ambient temperature and wind speed etc.

In this paper, the SCADA data of 24 wind turbines in 4 wind farms were used in the experiment. The overview of the 4 wind farms is shown in the TABLE I. These 4 wind farms are located in Jiangsu, Tianjin, Hubei, and Shanghai, China. Their data acquisition time is roughly the entire year of 2019, and they all contain 39 wind turbine variables. Among them, the wind farms in Jiangsu and Hubei are onshore wind farms, and the other two are offshore wind farms. The wind farms in Jiangsu, Tianjin and Hubei are equipped with 2.0MW wind turbines, while the Shanghai wind farm is equipped with 3.6MW wind turbines. The next experiment will not only verify the transfer capability of the proposed wJMAN, but also analyze the impact of wind farm types and wind turbine power on the transfer process.

2) Data Description

For convenience, the datasets from these 4 wind farms are called dataset A, dataset B, dataset C and dataset D respectively. This paper selects three fault states (hydraulic fault, converter fault and electric generator fault) and normal states of the collected SCADA data to verify the proposed wJMAN method. The amount of data in different states from the 4 datasets are shown in TABLE II. Since the vast majority of the collected SCADA data is normal data, according to the needs of the experiment, only a small part of them are randomly selected to participate in the experiment. As for the data of the other three fault states in the SCADA data, as shown in TABLE II, their data accounted for a very small proportion, so they were all used in the experiment.

TABLE I. OVERVIEW OF 4 WIND FARMS

Wind farms	Wind turbines	Acquisition time	Year of installation	Type of wind farm	Data size	Output power	Variable size
Jiangsu	#4#6#10#12 #15#23#24#26	2019/1/1 - 2019/11/25	2015	Onshore	334869	2.0MW	39
Tianjin	#1#3#7#13 #14#16#19	2019/1/1 - 2019/12/30	2016	Offshore	349003	2.0MW	39
Hubei	#2#6#15#19	2019/1/1 - 2019/11/20	2014	Onshore	167766	2.0MW	39
Shanghai	#1#2#3#4#7	2019/1/1 - 2019/9/23	2011	Offshore	190238	3.6MW	39

TABLE II. THE AMOUNT OF DATA OF 3 DATASETS IN THE 4 STATES

Wind farms	States	Proportion and the sum		Data size	
Jiangsu (Dataset A)	Normal	/	/	963	1291
	Hydraulic fault	0.047%		158	
	Converter fault	0.033%	0.098%	110	
	Ele-gen fault	0.018%		60	
Tianjin (Dataset B)	Normal	/	/	649	1192
	Hydraulic fault	0.046%		160	
	Converter fault	0.093%	0.156%	324	
	Ele-gen fault	0.017%		59	
Hubei (Dataset C)	Normal	/	/	646	905
	Hydraulic fault	0.020%		34	
	Converter fault	0.031%	0.154%	53	
	Ele-gen fault	0.103%		172	
Shanghai (Dataset D)	Normal	/	/	519	875
	Hydraulic fault	0.116%		221	
	Converter fault	0.025%	0.187%	47	
	Ele-gen fault	0.046%		88	

As shown in TABLE I, the installation years of the four wind farms are 2015, 2016, 2014 and 2011 respectively. Although they are not very new wind farms, the proportions of their three types of fault data also reflect the old and new levels to a certain extent, which is shown in TABLE II: for onshore wind farms (Jiangsu and Hubei), the newer Jiangsu wind farms have a lower fault ratio. ; For offshore wind farms (Tianjin and Shanghai), the proportion of faults in the newer Tianjin wind farm is lower than that of the Shanghai wind farm. Therefore, the collected data can be used to verify the proposed wJMAN for solving small samples of new wind farms.

B. Data Preprocessing

Comparison between the classification ability of different algorithms has some prerequisites. First, the data distribution must be balanced. That is, the number of categories in the data is roughly the same. Because in experiments based on unbalanced data, traditional classification algorithms that take the overall classification accuracy as the learning goal will pay too much attention to the majority class, which reduces the classification performance of the minority class samples. This is why most common machine learning algorithms do not work well with unbalanced datasets.

In order to deal with the unbalanced data, the data of the 4 states need be balanced. We adopted the synthetic minority oversampling technique (SMOTE) [27] method to balance the data. The SMOTE formula is as follows:

$$x_{new} = x + rand(0,1) * (x_i - x) \quad (9)$$

where x is any piece of data in a set of X , x_i is one of the k points closest to x , $x_i \in \{x_1, x_2, \dots, x_k\}$. $rand(0,1)$ is a random number generated from the interval $(0,1)$. x_{new} is a new piece of data based x .

Meanwhile, in order to reduce the influence between different variables, each variable is standardized by 0-1 normalization. The normalization formula is as follows:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (10)$$

where X represent a set of data, X_{max} is the maximum values in X , X_{min} is the and minimum values in X , and X_{norm} is the normative value of X .

TABLE III. STRUCTURE PARAMETERS OF WJMAN

Layers type	Activation function	Parameter size	Output size
Input	/	/	(39*3=13*9, 1)
Conv2D	ReLU	1-32; kernel=2x2; padding=0	(12*8, 32)
Conv2D	ReLU	32-128; kernel=3x3; padding=1	(12*8, 128)
MaxPool2D	/	Pool size=2; stride=2	(6*4, 128)
Conv2D	ReLU	128-256; kernel=3x3; padding=1	(6*4, 256)
MaxPool2D	/	Pool size=2; stride=2	(3*2, 256)
Flatten	/	/	(1536)
FC layer	ReLU	256*3*2-64; Dropout=0.2	(64)
FC layer	ReLU	64-16; Dropout=0.2	(16)
Classification layer	SoftMax	16-4	(4)

C. Model Structure and Training Hyperparameters

The proposed wJMAN model has been verified by many experiments, and the optimal structure obtained is shown in TABLE III. Among them, "Conv2D" indicates a two-dimensional convolutional layer, "MaxPool2D" indicates a two-dimensional maximum pooling layer, "Flatten" indicates that the input multi-dimensional data is compressed into a one-dimensional variable, "FC layer" indicates a fully connected layer, and "Classification layer" represents the final classification layer whose output is the prediction result of the source domain data. "padding=0" means to directly perform the convolution operation on the feature map; while "padding=1" means to fill the edges first, and then perform the convolution operation on the input feature map, so that the output feature map size is the same as the input feature map size. "stride" refers to the sliding stride of the receptive field, and the default value is "1". "Dropout" refers to the loss of part of the data immediately, which is a typical operation to prevent overfitting.

Because the CNN model requires a two-dimensional matrix as a structural attribute, and the collected SCADA data has only 39 variables, reconstruction into a two-dimensional matrix input will cause the input of the CNN to be too small. Therefore, we merge 3 consecutive data with the same label

together and then reconstruct these data into a two-dimensional matrix. And the input layer size is 39*3.

After many experimental verifications, we obtained the optimal training hyperparameters: learning rate, batchsize, and epoch were 0.005, 80, and 50, respectively. The optimizer adopts Adam optimizer.

D. Comparison Methods and Evaluations

In order to test the fault diagnosis effect of wJMAN, JMAN [18], common domain adaptive models JAN [26], DDC [24], DaNN [23] and a finetuning method, Finetuned CNN [14] are selected for comparison. The feature extraction modules and transfer strategies of these comparison methods and the proposed wJMAN are shown in TABLE IV.

Confusion matrix is a common evaluation index that can intuitively indicate the effect of classification model. Other common evaluation indicators of classification models, such as accuracy, precision, recall, F1-score, can be calculated by the confusion matrix. TABLE V shows the confusion matrix of the two-class problem for a certain state (normal state or a certain type of fault). Accuracy is the percentage of the sample that is predicted correctly. Precision is the percentage of the predicted samples which are correctly predicted. Recall is the percentage of actual samples are correctly predicted. F1-score is the

harmonic average of Precision and Recall. It also takes into account both the precision and recall of the classification model.

In this paper, accuracy and F1-score are used to evaluate the overall and specific category detection performance of these methods. The definitions of these evaluation indicators are as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (11)$$

$$Precision = \frac{TP}{TP+FP} \quad (12)$$

$$Recall = \frac{TP}{TP+FN} \quad (13)$$

$$F_1 - score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2*TP}{2*TP+FP+FN} \quad (14)$$

TABLE IV. FEATURE EXTRACTION MODULES AND TRANSFER STRATEGIES OF THE COMPARISON METHODS AND wJMAN

Methods	Feature extraction	Transfer strategy
wJMAN	CNN	JMMD-based weighted adaptation & Matching
JMAN	CNN	JMMD-based adaptation & Matching
JAN	CNN	JMMD-based adaptation
DDC	CNN	MMD-based adaptation
DaNN	NN	MMD-based adaptation
Finetuned CNN	CNN	Parameter transfer

TABLE V. CONFUSION MATRIX OF THE TWO-CLASS PROBLEM

Confusion matrix		Predicted class	
		Positive	Negative
True class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

E. Detail Analysis of the proposed wJMAN

1) Convergence Analysis

As shown in Fig. 7, the changes of the overall loss, classification loss and adaptive loss (JMMD loss) of the proposed wJMAN in the iterative learning process of transferring dataset A to dataset B are plotted for convergence analysis. At the same time, for the convenience of comparison, Fig. 8 shows the changes in the overall loss, classification loss and JMMD loss during the JMAN iteration process of transferring dataset A to dataset B.

For wJMAN, its overall loss is an adaptive weighted sum of classification loss and JMMD loss. In the early stage of

iterative learning, the classification loss has a larger proportion, which means that the classification loss needs to be optimized first, and the weight of the classification loss in the next iteration should be larger. Optimizing the classification loss first makes the overall loss drop faster. When the epoch is equal to 11, the classification loss is already smaller than the adaptive loss. In the subsequent iterative learning process, the classification loss and the adaptive loss tend to balance, and when the epoch is around 50, the network reaches convergence.

The overall loss of JMAN is half of the sum of the classification loss and JMMD loss, that is, the weight of both types of loss is 0.5. In the early stage of iterative learning, the classification loss accounts for a larger proportion, but the weight of the classification loss does not change. Therefore, compared with wJMAN, the overall loss of JMAN decreases slowly and smoothly. The classification loss is always larger than the adaptive loss, and the size of the two converges when the epoch is around 30. As the epoch increases, many ups and downs and glitches appear. When the epoch is 50, the network has not yet reached convergence.

Fig. 7. The overall loss, classification loss and JMMD loss of wJMAN in continuous iterative learning of transferring dataset A to dataset B

Fig. 8. The overall loss, classification loss and JMMD loss of JMAN in continuous iterative learning of transferring dataset A to dataset B

In general, compared to JMAN, the increased adaptive weight of wJMAN allows the model to preferentially optimize the worse loss (larger loss), making the model converge faster. But in the iterative learning process, the volatility of wJMAN is more prominent.

2) Adaptive Loss Weight Analysis

Fig. 9. The weights of the two classes of wJMAN with increasing iterations (transferring from dataset A to dataset B)

Fig. 10. The weights of the two classes of wJMAN with increasing iterations (transferring from dataset B to dataset A)

In order to further analyze the adaptive loss weights of wJMAN, we visualize the changes of the weights of the classification loss and the weights of the JMMD loss during the iterative learning process. Fig. 9 and Fig. 10 respectively show

the changes of the two types of weights of wJMAN as the iteration increases when dataset A is transferred to dataset B and dataset B is transferred to dataset A, respectively. From Fig. 9 and Fig. 10, we obtain the following observations:

a) As mentioned earlier, according to Equation (11) and Equation (12), the weight of the classification loss plus the weight of the JMMD loss is equal to 1. As shown in Fig. 9 and Fig. 10, this characteristic of the adaptive loss weight can be confirmed again. And the weight curve of the classification loss and the weight curve of the JMMD loss are axisymmetric about the line "weight=0.5".

b) For Fig. 9 and Fig. 10, at the beginning of the iterative learning of wJMAN, the classification loss greatly exceeds the JMMD loss, so the weight of the classification loss is also larger. As the epoch increases, the weight curve of the classification loss shows a certain downward trend, and the shape is burr, until it converges to 0.5 and gradually stabilizes. The difference between Fig. 9 and Fig. 10 is that the former converges and stabilizes after epoch reaches 13, while the latter converges and stabilizes after epoch reaches 25.

3) Feature Visualization

In order to clearly show the effect of wJMAN, taking the mutual transferring process of dataset A and dataset B as an example, the t-SNE visualization of the features extracted by wJMAN and the raw data is shown in the Fig. 11. It can be clearly found that the source domain and target domain obtained by wJMAN extraction have similar feature distributions, which shows that wJMAN can effectively construct an invariant feature space. More specifically, Fig. 11 (a) and Fig. 11 (c) respectively represent the raw data t-SNE visualization of the mutual transfer process of Dataset A and Dataset B. It can be clearly found that the distinction between different states of the same dataset is not obvious, and the original data distribution of the two datasets is quite different. As shown in Fig. 11 (b) and Fig. 11 (d), after wJMAN learning, not only the boundary of the different state features of the same dataset is obvious, but the features of the same state of the two datasets are highly overlapped.

Fig. 11. The t-SNE visualization of raw data and features extracted by wJMAN

4) Diagnostic Results

The detailed fault diagnosis results of wJMAN are shown in TABLE VI. Since the 4 wind farms are equipped with wind turbines with different capacities, the transfer process of the 2.0 MW wind farms is first analyzed. The proposed wJMAN method has excellent diagnostic accuracy, with an average accuracy exceeding 0.985. In the experiment of transferring dataset C to dataset B, the average precision even reached 1. For normal state and generator failure state, their F1 scores are slightly lower than those of hydraulic failure state and converter failure state.

For the transfer process between the 3.6MW wind farm and the 2.0MW wind farm, as shown in Fig. 10, it can be clearly seen that when the 3.6MW dataset D is used as the source domain, the diagnostic performance of wJMAN is better than when it is used as the target domain. For the three transfer processes with dataset D as the source domain, their accuracies are almost all equal to 1. The accuracies of the three transfer processes of dataset D as the target domain are also excellent, all exceeding 0.989.

Fig. 12. The detailed diagnosis result of the proposed wJMAN method

TABLE VI. THE DETAILED DIAGNOSIS RESULT OF THE PROPOSED METHOD

Transfer direction		Accuracy	F1-score (normal)	F1-score (hydraulic)	F1-score (converter)	F1-score (Ele-gen)
2.0MW ↓ 2.0MW	A→B	0.992 ± 0.008	0.984 ± 0.016	1.000 ± 0.000	0.996 ± 0.004	0.988 ± 0.013
	A→C	0.985 ± 0.015	0.971 ± 0.028	1.000 ± 0.000	1.000 ± 0.000	0.967 ± 0.031
	B→A	0.991 ± 0.007	0.982 ± 0.013	0.999 ± 0.002	0.999 ± 0.001	0.984 ± 0.015
	B→C	0.998 ± 0.002	0.996 ± 0.004	1.000 ± 0.000	1.000 ± 0.000	0.997 ± 0.004
	C→A	0.989 ± 0.008	0.987 ± 0.010	0.991 ± 0.004	0.998 ± 0.001	0.980 ± 0.016
	C→B	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
3.6MW ↓ 2.0MW	A→D	0.998 ± 0.002	1.000 ± 0.000	0.996 ± 0.004	1.000 ± 0.000	0.997 ± 0.004
	B→D	0.995 ± 0.001	0.999 ± 0.000	0.990 ± 0.000	0.990 ± 0.000	0.999 ± 0.001
	C→D	0.989 ± 0.001	1.000 ± 0.002	0.976 ± 0.002	0.994 ± 0.006	0.983 ± 0.007
	D→A	0.999 ± 0.001	0.999 ± 0.000	0.997 ± 0.002	1.000 ± 0.000	0.998 ± 0.003
	D→B	1.000 ± 0.000	1.000 ± 0.000	0.999 ± 0.001	0.999 ± 0.002	1.000 ± 0.000
	D→C	1.000 ± 0.000	1.000 ± 0.001	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000

F. Overall Results of wJMAN and the Comparison Methods

The Fig. 11 and TABLE VII show the average diagnostic accuracy and deviation of wJMAN and the competing methods. It is not difficult to find that wJMAN has the best diagnostic performance among all methods and its average diagnostic accuracy reached 0.994. Specifically, in the 12 transfer process experiments, wJMAN has the highest average diagnosis accuracy in 11 experiments, and its average diagnostic accuracy is tied with JMAN in 1 experiment. At the same time, the proposed wJMAN has a small bias, which also means that its diagnostic stability is high.

Among the comparison methods, JMAN ranks second in diagnostic performance with an average diagnostic accuracy of 0.854, and the fine-tuned CNN ranks third in diagnostic performance with an average diagnostic accuracy of 0.761. The diagnostic performance of JAN and DaNN is somewhat inferior, and the average diagnostic accuracy is between 0.5 and 0.6. The DDC has the worst diagnostic performance, with an average diagnostic accuracy of only 0.404 in the 12 experiments.

To reveal the role of subdomain matching and adaptive loss weights in the proposed wJMAN, these methods are compared in detail as follows:

1) wJMAN vs JMAN

In all 12 transfer processes, the average accuracies of wJMAN exceed that of JMAN by more than 0.1 in 7 transfer processes. In the 3 transfer processes, the average accuracies of wJMAN exceed that of JMAN by 0.01-0.1. In the remaining 2 transfer processes, the average accuracies of wJMAN and JMAN are almost the same. These illustrate that the proposed adaptive loss weights in the proposed wJMAN can help improve the diagnostic accuracy of the model.

2) wJMAN & JMAN vs Other comparison methods

Obviously, wJMAN and JMAN are the best and second-best models for fault diagnosis, respectively. The third best model for fault diagnosis is the fine-tuned CNN, the average diagnostic accuracy of wJMAN exceeds that of fine-tuned CNN by more than 0.2, and the average diagnostic accuracy of JMAN exceeds that of fine-tuned CNN by 0.09. The JMAN model only adds a sub-domain matching module on the basis of JAN, and its average diagnostic accuracy exceeds that of JAN by 0.27. All of the above shows that sub-domain matching can improve the diagnostic performance of the model.

TABLE VII. DIAGNOSTIC ACCURACY OF JMAN AND COMPARISON METHODS

Transfer Direction		Methods					
		wJMAN	JMAN	JAN	DDC	DaNN	Finetuned CNN
2.0MW ↕ 2.0MW	A→B	0.992±0.008	0.992±0.008	0.68±0.072	0.421±0.014	0.589±0.011	0.686±0.083
	A→C	0.985±0.015	0.833±0.052	0.905±0.051	0.516±0.016	0.662±0.008	0.869±0.003
	B→A	0.991±0.007	0.773±0.027	0.77±0.036	0.373±0.003	0.475±0.055	0.720±0.025
	B→C	0.998±0.002	0.789±0.094	0.705±0.024	0.578±0.003	0.675±0.024	0.889±0.020
	C→A	0.989±0.008	0.876±0.022	0.653±0.086	0.328±0.013	0.464±0.005	0.633±0.146
	C→B	1.000±0.000	0.957±0.012	0.746±0.095	0.512±0.038	0.736±0.018	0.800±0.035
2.0MW ↕ 3.6MW	A→D	0.998±0.002	0.939±0.001	0.458±0.106	0.499±0.067	0.550±0.031	0.741±0.099
	B→D	0.995±0.001	0.841±0.012	0.268±0.043	0.221±0.072	0.24±0.004	0.777±0.088
	C→D	0.989±0.001	0.987±0.002	0.439±0.158	0.43±0.021	0.558±0.052	0.751±0.125
	D→A	0.999±0.001	0.619±0.097	0.367±0.034	0.389±0.075	0.377±0.030	0.800±0.055
	D→B	1.000±0.000	0.700±0.064	0.371±0.121	0.19±0.022	0.527±0.064	0.621±0.007
	D→C	1.000±0.000	0.940±0.020	0.563±0.001	0.391±0.105	0.497±0.033	0.843±0.013
Average		0.994	0.854	0.577	0.404	0.529	0.761

Fig. 13. Diagnosis accuracy histogram of wJMAN and comparison methods

G. Overall Results of wJMAN and the Comparison Methods

The Fig. 11 and TABLE VII show the average diagnostic accuracy and deviation of wJMAN and the competing methods. It is not difficult to find that wJMAN has the best diagnostic performance among all methods and its average diagnostic accuracy reached 0.994. Specifically, in the 12 transfer process experiments, wJMAN has the highest average diagnosis accuracy in 11 experiments, and its average diagnostic accuracy is tied with JMAN in 1 experiment. At the same time, the proposed wJMAN has a small bias, which also means that its diagnostic stability is high.

Among the comparison methods, JMAN ranks second in diagnostic performance with an average diagnostic accuracy of 0.854, and the fine-tuned CNN ranks third in diagnostic performance with an average diagnostic accuracy of 0.761. The diagnostic performance of JAN and DaNN is somewhat inferior, and the average diagnostic accuracy is between 0.5 and 0.6. The DDC has the worst diagnostic performance, with an average diagnostic accuracy of only 0.404 in the 12 experiments.

To reveal the role of subdomain matching and adaptive loss weights in the proposed wJMAN, these methods are compared in detail as follows:

1) wJMAN vs JMAN

In all 12 transfer processes, the average accuracies of wJMAN exceed that of JMAN by more than 0.1 in 7 transfer processes. In the 3 transfer processes, the average accuracies of wJMAN exceed that of JMAN by 0.01-0.1. In the remaining 2 transfer processes, the average accuracies of wJMAN and JMAN are almost the same. These illustrate that the proposed

adaptive loss weights in the proposed wJMAN can help improve the diagnostic accuracy of the model.

2) wJMAN & JMAN vs Other comparison methods

Obviously, wJMAN and JMAN are the best and second-best models for fault diagnosis, respectively. The third best model for fault diagnosis is the fine-tuned CNN, the average diagnostic accuracy of wJMAN exceeds that of fine-tuned CNN by more than 0.2, and the average diagnostic accuracy of JMAN exceeds that of fine-tuned CNN by 0.09. The JMAN model only adds a sub-domain matching module on the basis of JAN, and its average diagnostic accuracy exceeds that of JAN by 0.27. All of the above shows that sub-domain matching can improve the diagnostic performance of the model.

V. CONCLUSION

This paper studies the problem of insufficient labeled data for wind turbines in new wind farms, and proposes a new transfer learning method: weighted joint matching adaptive network. On the basis of JMAN, wJMAN retains subdomain matching, and innovatively proposes a method of adaptive loss weight to adaptively weight the classification loss and adaptive loss in the overall loss. The source domain and target domain are aligned by sub-domain matching and minimizing the joint maximum mean discrepancy, and then the invariant feature space of the two domains is obtained. This method can transfer information from other existing wind farms to the target new wind farm by constructing the invariant feature space of the target new wind farm and other wind farms. Its performance is proved by real data from 4 wind turbines.

VI. DECLARATIONS

A. Acknowledgment

Not applicable.

B. Authors' contribution

GZ was in charge of the whole trial and wrote the manuscript; YL gave some advices on manuscript and approved the final manuscript.

C. Funding

This work was supported by the National Natural Science Foundation of China under Grant 72072114.

D. Competing interests

The authors declare that there are no competing interests regarding the publication of this paper.

E. Author details

School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200000, China

REFERENCES

- [1] G. W. E. Council, Global Wind Report 2018. No. 1, Global Wind Energy Council, 2019.
- [2] Y. Ding, Data Science for Wind Energy. No. 1, CRC Press, 2019.
- [3] J. Fu, J. C. Chu, P. Guo, and Z. Y. Chen, "Condition monitoring of wind turbine gearbox bearing based on deep learning model," IEEE Access, vol. 7, no. 99, pp. 57078–57087, 2019.
- [4] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," IEEE Transactions on Industrial Electronics, vol. 65, no. 99, pp. 5990–5998, 2018.
- [5] C. Lu, Z.-Y. Wang, W. L. Qin, and J. Ma, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification," Signal Processing, vol. 130, pp. 377–388, 2017.
- [6] X. Fu, H. Chen, G. Zhang, and T. Tao, "A new point anomaly detection method about aero engine based on deep learning," In 2018 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), pp. 176–181, 2018.
- [7] H. Liu, J. Zhou, Y. Zheng, W. Jiang, and Y. Zhang, "Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders," Isa Transactions, vol. 77, pp. 167–178, 2018.
- [8] P. Tamilselvan and P. Wang, "Failure diagnosis using deep belief learning based health state classification," Reliability Engineering & System Safety, vol. 115, pp. 124–135, 2013.
- [9] L. Xiang, et al. "Fault detection of wind turbine based on SCADA data analysis using CNN and LSTM with attention mechanism." Measurement 175.8:109094, 2021.
- [10] J. Lei, C. Liu, and D. Jiang, "Fault diagnosis of wind turbine based on Long Short-term memory networks." Renewable Energy, 133:422–432, 2019.
- [11] M. A. Pliego, M. García, P. Pinar, and R. H. Diego, "A survey of artificial neural network in wind energy systems," Applied Energy, vol. 228, pp. 1822–1836, 2018.
- [12] Y. Lei, B. Yang, X. Jiang, et al. Applications of machine learning to machine fault diagnosis: A review and roadmap[J]. Mechanical Systems and Signal Processing, 138: 106587, 2020.
- [13] X. Yang, et al. "Gas path fault diagnosis for gas turbine group based on deep transfer learning." Measurement, 109631, 2021.
- [14] Y. Li, W. Jiang, G. Zhang, and L. Shu, "Wind turbine fault diagnosis based on transfer learning and convolutional autoencoder with small-scale data," Renewable Energy, vol. 171, no. 3, 2021.
- [15] Y. Liao, et al. "Dynamic Distribution Adaptation Based Transfer Network for Cross Domain Bearing Fault Diagnosis." Chinese Journal of Mechanical Engineering, 34.1: 1–10, 2021.
- [16] L. Wen, L. Gao, and X. Li, "A new deep transfer learning based on sparse auto-encoder for fault diagnosis," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 1, pp. 136–144, 2019.
- [17] X. Li, H. Jiang, R. Wang, and M. Niu, "Rolling bearing fault diagnosis using optimal ensemble deep transfer network," Knowledge-Based Systems, vol. 213, p. 106695, 2020.
- [18] G. Zhang, Y. Li, "A Fault Diagnosis Method for Wind Turbines of New Wind Farm Based on Joint Matching Adaptive Network," 2021 IEEE International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), pp. 240–245, IEEE, 2021.
- [19] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," Bioinformatics (Oxford, England), vol. 22, no. 14, pp. 49–57, 2006.
- [20] P. S. J. T. I. W, K. J. T, and Y. Q, "Domain adaptation via transfer component analysis," IEEE Transactions on Neural Networks, vol. 22, no. 2, pp. 199–210, 2011.
- [21] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric tri-training for unsupervised domain adaptation," in International Conference on Machine Learning, pp. 2988–2997, PMLR, 2017.
- [22] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in 2013 IEEE International Conference on Computer Vision, pp. 2200–2207, 2014.
- [23] M. Ghifary, W. B. Kleijn, and M. Zhang, "Domain adaptive neural networks for object recognition," in Pacific Rim International Conference on Artificial Intelligence, pp. 898–904, Springer, 2014.
- [24] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," Computer Science, 2014.
- [25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems 25, pp. 1097–1105, 2012.
- [26] M. Long and J. Wang, "Learning transferable features with deep adaptation networks," Proceedings of the 32nd International Conference on Machine Learning, PMLR, pp. 37:97–105, 2015.
- [27] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, no. 1, pp. 321–357, 2011.