EURASIP Journal on Advances in Signal Processing

RESEARCH **Open Access**

Unsupervised domain adaptive bearing fault diagnosis based on maximum domain discrepancy

Cuixiang Wang[1], Shengkai Wu[1,2] and Xing Shao[1*]

*Correspondence: shaoxing269@sina.com

[1] School of Information Engineering, Yancheng Institute of Technology, Yancheng 224000, Jiangsu, China
[2] School of Mechanical Engineering, Yancheng Institute of Technology, Yancheng 224000, Jiangsu, China

Abstract

In the existing domain adaptation-based bearing fault diagnosis methods, the data difference between the source domain and the target domain is not obvious. Besides, parameters of target domain feature extractor gradually approach that of source domain feature extractor to cheat discriminator which results in similar feature distribution of source domain and target domain. These issues make it difficult for the domain adaptation-based bearing fault diagnosis methods to achieve satisfactory performance. An unsupervised domain adaptive bearing fault diagnosis method based on maximum domain discrepancy (UDA-BFD-MDD) is proposed in this paper. In UDA-BFD-MDD, maximum domain discrepancy is exploited to maximize the feature difference between the source domain and target domain, while the output feature of target domain feature extractor can cheat the discriminator. The performance of UDA-BFD-MDD is verified through comprehensive experiments using the bearing dataset of Case Western Reserve University. The experimental results demonstrate that UDA-BFD-MDD is more stable during training process and can achieve higher accuracy rate.

1 Introduction

In the field of industrial production, maintaining long-term reliable operation of machinery is crucial for enterprises [1]. If mechanical equipment malfunctions, it may lead to production stoppage and even personal injury, resulting in huge losses for enterprises. However, with the development of intelligent manufacturing, mechanical equipment is becoming increasingly complex. In addition, the working environment of mechanical equipment is usually harsh or its working load is heavy, which makes the machinery maintenance complex and difficult [2]. Besides, according to statistics, more than half of the failures of mechanical equipment are caused by the failure of rotating bearing parts due to its heavy load and long-time high-speed rotation [3]. Therefore, it is necessary and important to conduct predictive monitoring and maintenance of rotating bearings to ensure the long-term reliable operation of bearings and mechanical equipment [4]. When the faults are found, bearing can be promptly inspected and repaired to avoid serious accidents and long-term downtime for maintenance.

In predictive monitoring, sensors are usually installed on mechanical equipment to monitor and record the status of mechanical equipment [5, 6]. And the bearing fault diagnosis is conducted using the collected data, including vibration signals, force signals and audio signals. First, features are extracted from the collected data. Then, the traditional machine learning methods, e.g., support vector machine (SVM), decision tree (DT), artificial neural network (ANN), are exploited to classify the status of bearings according to the extracted features. However, the features extraction and features selection for classification depend on expertise. In addition, traditional machine learning methods as shallow learning methods have limitations in learning ability [7].

With the development of the Internet of Things (IoT) [8] and cloud computing technology [9], a large amount of data about the operating status of machinery can be recorded and saved, which provides the possibility for mechanical fault diagnosis based on deep learning. Deep learning demonstrates powerful feature extraction ability, which makes it possible to explore the inherent characteristics of bearing signals and realize the fault diagnosis of bearings [10–12].

Although deep learning has been widely used in bearing fault diagnosis, the following problems still exist when it is applied in real industrial scenarios [13]: (1) When deep learning is exploited for bearing fault detection, the accuracy of fault diagnosis and model learning is positively correlated with the amount of collected monitoring data. However, collecting large amount of data in real industrial scenarios is difficult and expensive. (2) The working environment of bearings in real industrial scenarios varies under different working conditions. When the working environment changes, the deep learning model established under a certain working condition cannot be directly used to complete the bearing fault diagnosis task under the new working condition. Therefore, a new deep learning model needs to be trained using the new data collected from the new working environment.

In view of the above problems, transfer learning [14] is introduced to complete the bearing fault diagnosis under variable working conditions. Transfer learning takes the bearing fault data collected in one working condition as source domain and the data from another working condition as target domain. Transfer learning methods can find the common space of the interrelated features of the source domain data and the target domain data [15]. Therefore, the model using transfer learning under one working condition can be transferred to complete the bearing fault diagnosis under different conditions. With the development of transfer learning, domain adaptation, as an important transfer learning method, is widely used in bearing fault diagnosis. Li et al. [16] proposed to minimize the maximum mean discrepancy (MMD) between two domains at multiple layers and adapted the learned representations from the source domain to be applied in the target domain. Yang et al. [17] proposed a multi-layer domain adaptation (MLDA) method, which adopts multi-kernel maximum mean discrepancy (MK-MMD) and pseudo-label learning in multi-layer and considers marginal distribution and conditional distribution. MLDA can simultaneously diagnose multi-scale composite failures and single failures. Besides, domain adaptation methods based on adversarial methods are also used in bearing fault diagnosis. Liu et al. [18] proposed a deep adversarial domain adaptation (DADA) model for rolling bearing fault diagnosis, which combined deep stacked autoencoder (DSAE) with representative feature learning for dimension reduction to effectively acquire fault features. Wang [19] adopted the adversarial method based on Wasserstein distance to achieve data-level alignment, further minimized the discrepancy in the spatial distribution of class-level features through three sets of losses and proposed a three-group loss-guided adversarial domain adaptation method (TLADA) for bearing faults diagnosis.

However, adversarial domain adaptive methods have following problems that affect the diagnosis effect: (1) When the training of the target domain feature extractor is completed, the training is not stopped in time. In order to confuse the discriminator, the target domain-aligned features may map to the wrong class. (2) When the bearing fault data in the source domain and the target domain are indistinguishable, the weights of the feature extractor of target domain will gradually approach that of source domain, which impacts the model training effectiveness.

Aiming at the above problems, e.g., poor cross-working condition bearing fault diagnosis performance, difficulty in obtaining labeled data in the target domain and training of adversarial domain adaptive methods, this paper proposes an unsupervised domain adaptive bearing fault diagnosis method based on maximum domain discrepancy (UDA-BFD-MDD).

As shown in the left side of Fig. 1, the generator of the general domain adaptation methods generate features that confuse the discriminator to make the two distributions similar. The boundary relationship between different classification labels is not considered. UDA-BFD-MDD learns to maximize the discrepancy between the source and target domain features. On the one hand, UDA-BFD-MDD makes the features output by the target domain feature extractor as confusing as possible to the discriminator, making the distributions of the two domains similar. On the other hand, target domain extractor in UDA-BFD-MDD learns the difference within domain, so that the boundary between the source domain and target domain data in different label classifications is widened as shown in right part of Fig. 1, and the potential features of the source domain data are learned as much as possible.

The contributions of UDA-BFD-MDD are as follows: (1) In UDA-BFD-MDD, there is no need to label the target domain data under new working condition, which avoids the costs of data labeling and improves diagnostic efficiency. (2) Reversing labels is exploited, which has a larger gradient in back propagation algorithm, and quickly aligns the features of source domain and target domain. And UDA-BFD-MDD can still diagnose bearing fault even in the case of a small number of samples. (3) To ensure the target domain feature extractor obtains positive transfer as much as possible in training,

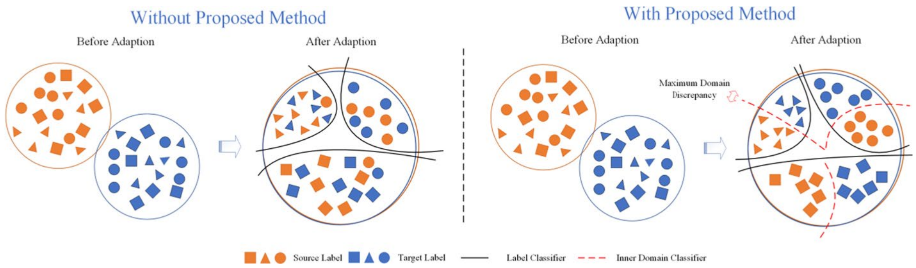


Fig. 1 Comparison of with and without maximum domain discrepancy methods

maximum domain discrepancy method is exploited. (4) The overall structure of UDA-BFD-MDD is light, and its calculation is simple. Compared with other domain adaptive methods, UDA-BFD-MDD can still achieve high bearing fault diagnosis accuracy and be applied in real industrial scenarios. To validate the performance of UDA-BFD-MDD, experiments are conducted on the Case Western Reserve University (CWRU) bearing dataset. The experimental results demonstrate that UDA-BFD-MDD can achieve stable and higher accuracy than other domain adaptive models under varied working conditions.

2 Relate work

Domain adaptation is considered as a special transfer learning method [20]. In literature [21], Pan et al. pointed out that transfer learning is the transfer of knowledge from a source domain to a different target domain, or from a source task to a different target task. And changes in feature space and marginal probability distributions may cause domain changes. Besides, Pan et al. proposed a transfer component analysis (TCA) to learn feature representation across domains in a high-dimensional regenerated kernel Hilbert space, where data from different domains are close to each other. Therefore, using TCA the standard machine learning methods can be applied in the spanned Hilbert space on different domains. Inspired by the theory on domain adaptation, Yaroslav et al. [22] proposed domain-adversarial neural networks (DANN), which is based on the neural network architecture and learns features from both labeled source domain data and unlabeled target domain data.

In domain adaptation methods, the task and feature space of the source and target domains need to be the same, while the marginal distribution of the two domains can be different [23]. Most domain adaptation methods are domain-invariant feature learning. If the feature representation of source domain and the target domain are consistent and the marginal distribution of labels is not very different, the features can be extracted from data through the neural network, and then, the features can be aligned to create an invariant domain with the same feature distribution. The invariant domain does not distinguish whether the data are from the source domain or the target domain as long as the feature representations of two domains are consistent. Therefore, the model established by learning the invariant domain features on the source domain can also be well generalized to the target domain. The flow of domain-invariant feature learning is shown in Fig. 2.

The main difference of existing domain-invariant feature learning methods is the domain alignment method. One of the commonly used alignment methods is to minimize the distance of the distribution. For example, maximum domain discrepancy (MMD) is proposed in [24] to calculate the average value of the difference to judge whether the distribution of the two domains in Hilbert space is the same. Besides, MMD is also used in deep adaptation network (DAN) [25] and deep convolutional transfer learning network (DCTLN) [26]. Sun et al. [27] proposed CORAL (Correlation Alignment), which is similar to MMD, but its value is obtained by computing the covariance of the source and target domain features. Another alignment approach is adversarial approach, whose alignment component consists of a domain discriminator. It draws on the idea of generative adversarial networks (GAN) [28], which

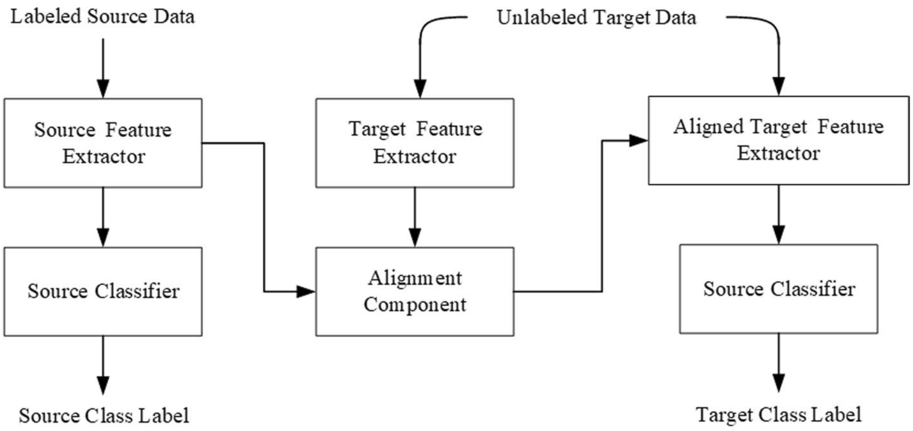


Fig. 2 Flowchart of domain-invariant feature learning

generates data through two network confrontation game. A value function in [28] is defined to quantify the game process of the two networks:

$$\min_G \max_D V(D,G) = E_{x\sim p_{\text{data}}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z)))] \quad (1)$$

In the formula (1), x is the real data, $p_z(z)$ is the input noise, $D(x)$ is the discriminator, and $G(z)$ is the generator. The goal of the value function $V(D,G)$ is to find the parameters so that the discriminator can correctly distinguish the real data x and the generated data $G(z)$, and at the same time find the parameters to minimize the value of $\log(1-D(G(z)))$. $D(G(z))$ is 0 when the discriminator correctly distinguishes the real input. When the discriminator treats the data generated by the generator as the real input, $D(G(z))$ is 1. Thus, the discriminator tries to learn to correctly classify the input as true or false, while the generator tries to trick the discriminator into thinking that the output it generates is real.

The adversarial alignment method in domain adaptation is modeled on the discriminator used in GAN to distinguish between true and generated data, and a domain discriminator is proposed to distinguish whether the data belong to the source domain or the target domain. The feature extractor characterizes the data, and the domain discriminator identifies which domain the data belongs to. The two networks are updated alternately. When the data cannot be correctly distinguished from the discriminator, the effect of the feature extractor is optimal. Ajakan et al. [29] added a gradient inversion layer to the feature extractor and domain discriminator during back propagation. In [30], a novel generalized adversarial adaptation framework is proposed. In addition, adversarial discriminative domain adaptation (ADDA) proposed by Tzeng et al. [30] uses the label inversion method to make the features of the target domain close to that of source domain. Furthermore, the deep domain confusion method (DDC) [31] uses an adaptive domain and confusion loss to learn domain-invariant representations.

Existing domain adaptation-based methods focus on the feature alignment. However, the data with same label from different domain after adaptation are not distinguished clearly. The target domain-aligned features may result in wrong map. And the weights of feature extractor of target domain are similar to that of source domain. This paper proposes an unsupervised domain adaptive bearing fault diagnosis method based on maximum domain discrepancy (UDA-BFD-MDD). UDA-BFD-MDD exploits maximum domain discrepancy to maximize the feature difference between the source domain and target domain. Comprehensive experiments on CWRU datasets confirms the performance of UDA-BFD-MDD.

3 UDA-BFD-MDD model

The UDA-BFD-MDD model consists of three parts, namely the pre-training module, the transfer module and the verification module. The structure of the UDA-BFD-MDD model is shown in Fig. 3. The pre-training module is to train source feature extractor and source classifier based on the labeled data from source domain. After pre-training module, the domain adaptation module is exploited, in which adversarial thinking is used to alignment the feature output of source feature extractor and target feature extractor. After the domain adaptation module, the unlabeled target data can be extracted feature using target feature extractor and then classified using source classifier as shown in verification module.

3.1 Pre-training module

In the pre-training module, in order to build a model for a specific task, it is necessary to train the source domain feature extractor and the source domain classifier to correctly classify the source domain sample data. The structure of the source domain feature extractor in the module is shown in Fig. 4.

The entire feature extractor consists of three feature extraction layers and one fully connected layer. Each feature extraction layer includes convolutional layers, pooling layers and activation functions. The data are first extracted through the convolution layer, and then, the extracted features are filtered through the pooling layer. The pooling method chosen is the max pooling method. Finally, the result is output through an activation function (ReLU). The structure output by the feature extractor is fed into the classifier to complete the data classification task on the source domain. The structure of the source domain classifier is shown in Fig. 5. In order to prevent over-fitting, the result

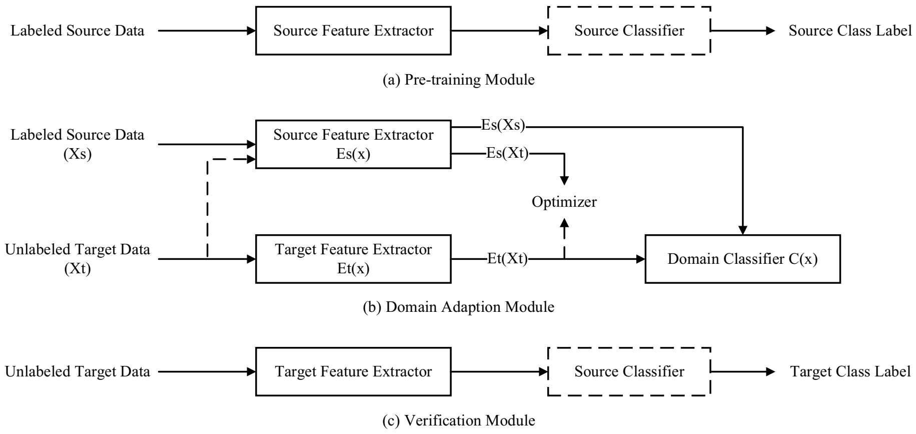


Fig. 3 Structure of UDA-BFD-MDD model

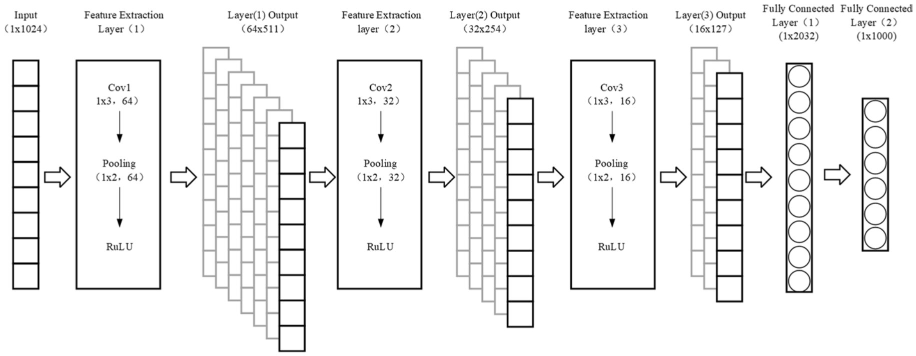


Fig. 4 Structure of source domain feature extractor

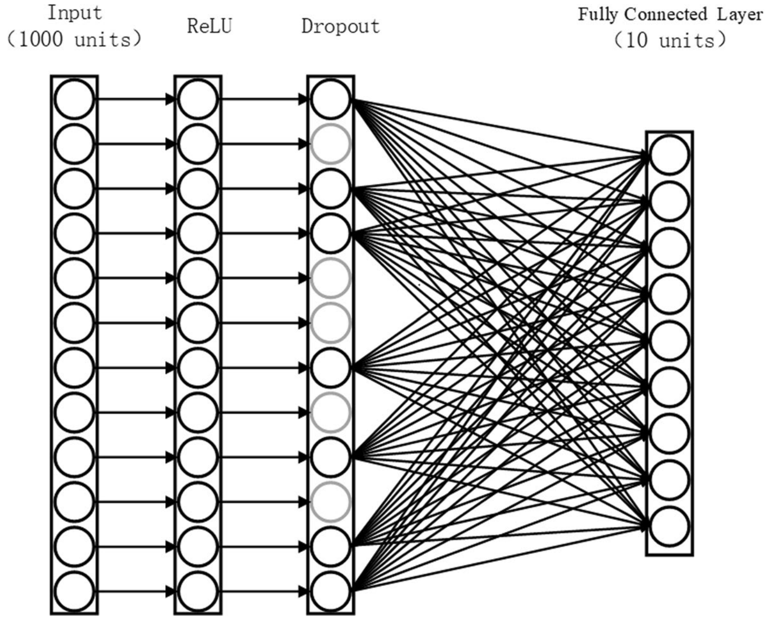


Fig. 5 Structure of source domain classifier

of feature extractor is input into the dropout layer before the fully connected layer, and a batch of neural network units is selected in the dropout layer so that it does not participate in the next parameter update [32]. Finally, two fully connected layers are used to complete the bearing fault classification.

3.2 Domain adaptation module

The model established in the source domain can be transferred to the target domain through the domain adaptation module to complete the bearing fault diagnosis task under different working conditions. The domain adaptation module consists of a source domain feature extractor, a target domain feature extractor, an optimizer and a domain discriminator. The source domain feature extractor is in the pre-train module, which has been trained with the source domain data, and can extract the basic features of the source domain data. The structure of the target domain feature extractor is the same as that of the source domain feature extractor, and at the beginning of the transfer, the weight of the target domain feature extractor is the same as the source domain, and then its weights are updated by the optimizer and the domain discriminator, so that the target domain features The extractor can also perform feature extraction tasks on the target domain data. The optimizer is aimed at the situation where the weights of the source and target domain feature extractors are convergent when some data of the two domains are highly similar, and the L1 distance is introduced to measure the source domain feature extractor and the target domain extractor to extract features from the same target domain data. To ensure that when the target domain feature is close to the source domain feature, the weight of the target domain feature extractor will not be the same as that of the source domain. The domain discriminator consists of three fully connected layers and a LogSoftmax layer. The entire domain discriminator network is shown in Fig. 6.

The data of the feature extractor are divided into two types through the three fully connected layers, and then, the two types of data are subjected to the LogSoftmax operation. Compared with the Softmax operation, the LogSoftmax operation performs a logarithmic operation on the value of Softmax. On the one hand, the logarithmic function will not overflow when derivation, and on the other hand, it can speed up the back propagation speed and improve the operation efficiency. During transfer, in order to retain the information learned on the source domain, the learned source domain feature extraction network is used as the initial network of the target domain feature extractor. The extracted features are distinguished from the source domain or the target domain by the domain discriminator. When the domain discriminator fails to discriminate correctly, the feature extractor of the target domain completes the transfer and optimization.

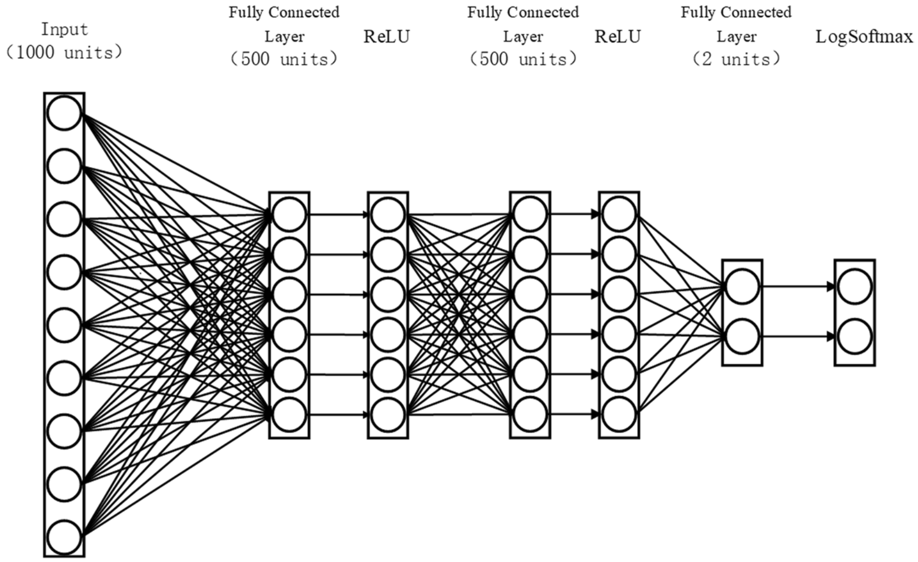

Input
(1000 units)
Fully Connected Layer (500 units)
ReLU
Fully Connected Layer (500 units)
ReLU
Fully Connected Layer (2 units)
LogSoftmax

Fig. 6 Domain discriminator network

3.3 Verification module

The verification module consists of a target domain feature extractor and a source domain classifier. After the domain adaptation module, the source domain feature extractor and target domain feature extractor have been alignment. Then, the data of the target domain are input into the target domain feature extraction, whose result is input into the source domain classifier trained by the data from source domain. Although the target domain data have no label and there is no new model is trained for the target domain data, the source domain classifier can classify the target domain data based on the feature alignment of source and target feature extractors. The detailed training steps are discussed in Sect. 3.4.

3.4 Training and optimization of UDA-BFD-MDD

The training part of UDA-BFD-MDD is divided into five parts, namely source domain feature extractor, source domain classifier, target domain feature extractor, optimizer and domain discriminator. The optimization process is divided into 4 steps as follows.

Step 1 Train source domain feature extractor and source domain classifier. The source domain data X_S is first input into the source domain feature extractor E_S, then the result of source domain feature extractor is input into the source domain classifier C_S, and the bearing faults are classified according to the label Y_S of the source domain data. Step 1 uses a standard loss function for training and optimization. The loss function is as follows:

$$\min_{E_s, C_s} L_{\text{cls}}(X_S, Y_S) = E_{(x_S, y_S) \sim (X_S, Y_S)} - \sum_{k=1}^{K} 1_{[k=y_S]} \log C_S(E_S(X_S)) \tag{2}$$

Step 2 Train the target domain feature extractor and optimizer. The target domain data X_T is input into the target domain feature extractor E_T for feature extraction. In order to make the features output by the target domain feature extractor similar to the source domain, the extracted features are input into the domain discriminator and trick the domain discriminator to misidentify the features as extracted from the source domain. This is used to measure and narrow the feature difference between the target domain and the source domain. At the same time, in order to ensure that the fault diagnosis task can continue to be completed even when the data in the source and target domains are similar, an optimizer is used to measure the difference loss of the two feature extractors. Combining the two, the overall loss function of step (2) can be obtained as follows:

$$\begin{aligned} \min_{E_T} \max_{Opt} V(D, E_T) &= \min_{E_T} \max_{\text{Opt}} L_{E_T}(X_t) + L_{opt}(X_s, X_t) \\ &= -E_{x_t \sim X_t}[\log D(E_t(x_t))] \\ &\quad + E_{(x_s, x_t) \sim (X_s, X_t)} \frac{\lambda}{n} \sum_{n=1}^{n} |E_t(x_{sn}) - E_t(x_{tn})| \end{aligned} \tag{3}$$

Step 3 Train the domain discriminator. When the target domain feature extractor cheats the domain discriminator in order to narrow the difference between the input features and the source domain, the domain discriminator also needs to be continuously optimized to distinguish as much as possible whether the features are extracted from the source domain or the target domain. In the continuous confrontation between the feature extractor and the domain discriminator, the feature extractor is continuously optimized. The loss function of the domain discriminator is as follows:

$$\min_{D}(X_s, X_t, E_s, E_t) = -E_{x_s \sim X_s}[\log D(E_s(x_s))] - E_{x_t \sim X_t}[\log D(E_t(x_t))] \tag{4}$$

Step 4 Combine the optimized target domain feature extractor with the source domain classifier to complete the fault diagnosis task. After several iterations of optimization in steps (2) and (3), the features of the target domain data extracted by the target domain feature extractor are as similar as possible to those of the source domain. These features are then input into the source domain classifier to complete the fault diagnosis of the new working condition.

3.5 Theoretical analysis of UDA-BFD-MDD

In the training and optimization process of UDA-BFD-MDD, the key step is the Step (2) which trains the target domain feature extractor and optimizer. The result of Step (2) will be a good target domain feature extractor and a good discriminator which make the discriminator misidentify the feature from the source domain or the target domain. Therefore, the result of Step (2) influences the performance of UDA-BFD-MDD directly. To facilitate understanding of the key Step (2), the theoretical analysis is presented in this section.

Given any target domain feature extractor E_T, the training for discriminator D is to minimize the quantity $V(D,E_T)$.

$$V(D, E_T) = -\int\limits_{X_T} p_{X_T} \log(D(E_T(X_T)))\mathrm{d}X_T + \iint\limits_{X_T,X_S} p_{X_T} p_{X_S} \frac{\lambda}{n} \sum_{n=1}^{N} |E_T(X_{Sn}) - E_T(X_{Tn})| \mathrm{d}X_T \mathrm{d}X_S \tag{5}$$

The training objective for D can be interpreted as maximizing the log-likelihood for estimating the conditional probability $P(Y=y|x)$, where Y indicates whether x comes from E_t or E_s. When the feature output of source and target feature extractor, i.e., $E_T(X_{Sn})$ is equal to $E_T(X_{Tn})$, the formula (5) can be reformulated as:

$$C(X_T) = \min_{X_T} V(D, E_T) = -E_{Xt \sim X_T}(\log D(E_T(X_T))) \tag{6}$$

4 Performance evaluation

In order to verify the performance of UDA-BFD-MDD on bearing fault diagnosis under different working conditions, comprehensive experiments are conducted using the rolling bearing data set of Case Western Reserve University (CWRU).

To obtain the bearing data of CWRU data set, EDM was used to implant faults ranging from 0.007 inches to 0.04 inches in diameter in various locations on the motor's bearings. Then, these faulty bearings were reinstalled into the motor. And the vibration data of the bearing were recorded when the motor works under different working conditions, which constituted the CWRU data set.

Table 1 CWRU dataset bearing failure classification labels

Label	Bearing working state	Fault damage location	Fault damage length
0	Fault	Ball	0.007
1	Fault	Ball	0.014
2	Fault	Ball	0.021
3	Fault	Inner ring	0.007
4	Fault	Inner ring	0.014
5	Fault	Inner ring	0.021
6	Fault	Outer ring	0.007
7	Fault	Outer ring	0.014
8	Fault	Outer ring	0.021
9	Normal	Normal	Normal

The bearing data selected for the experiments is the bearing data collected at the drive end using an accelerometer for a deep groove ball bearing of type 6205-2RS JEM SKF. According to the difference of horsepower, the data set is divided into 4 different working conditions: 0 horsepower, 1 horsepower, 2 horsepower and 3 horsepower, and the corresponding labels are 0, 1, 2, 3. And the sampling speed is 12,000 samples per second. The bearing data under each working condition are composed of bearing working signals of 9 fault states and 1 normal state, as shown in Table 1. Since the outer ring fault is a static fault, the relative position of the fault and bearing load has a direct impact on the data, and the outer ring fault position in Table 1 is the position orthogonal to the load. The labels in Table 1 are corresponding to different bearing faults.

4.1 Comparison of direct transfer and domain adaptive transfer

In order to verify the effect of adaptive transfer, the results of transfer from each working condition to other working conditions are compared. The result of direct transfer is obtained by directly transferring the model trained on the source domain in the pre-training process to the target domain, and adaptive transfer is the transfer of the

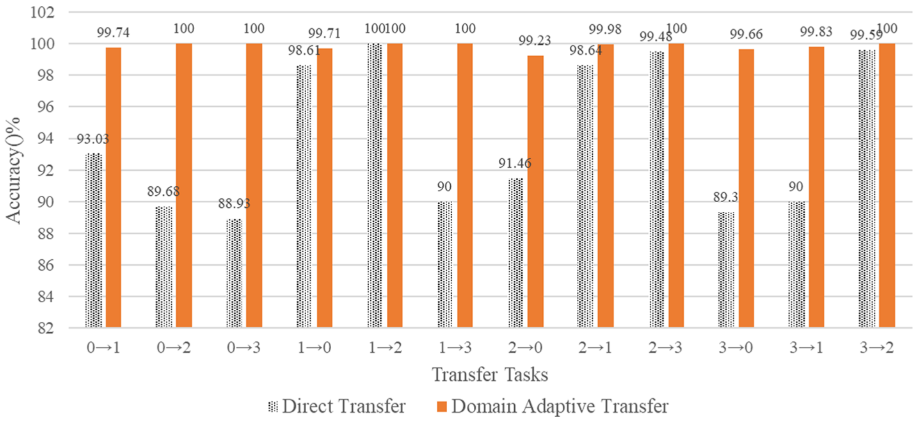


Fig. 7 Accuracy comparison of direct transfer and domain adaptive transfer

adaptive model after passing through the transfer module to the target domain. In the experiment, a number of signal data with a length of 2048 are randomly selected and input as samples after fast Fourier transform. The model is trained with 32 samples each time, and the loss function selects the cross entropy loss and uses Adam to optimize the model. The learning rate of Adam in the feature extractor and domain discriminator is set to 0.00001, and β_1 and β_2 are 0.5 and 0.9, respectively. The parameter of the optimizer is 0.1. In order to ensure that the result is not affected by randomness, final results are the average of 10 experimental results, and the results of the are shown in Fig. 7.

As shown in Fig. 7, the domain adaptive transfer in this paper is almost always better than that of direct transfer. The average accuracy rate of all domain adaptive transfer tasks is higher than 99%, and the average accuracy rate of 6 transfer tasks can reach 100%. It shows that when these 6 transfer tasks are transferred, they can fully adapt to the bearing fault diagnosis task of the new working condition. When transferring from working condition 2 to 0, the average accuracy rate of domain adaptive transfer is only 99.23. In the experiment, it is found that the highest accuracy rate of adaptive transfer from 2 to 0 can reach 100%, and the result fluctuates greatly, resulting in a low average accuracy rate. Overall, the final adaptive transfer results have nothing to do with the direct transfer results. Moreover, the results transferring from a low-horsepower condition to a high-horsepower condition are better than that of transferring from a high-horsepower condition to a low-horsepower condition.

4.2 Comparison with other transfer methods

In order to verify the performance of UDA-BFD-MDD, comparative experiments of UDA-BFD-MDD with other transfer methods were conducted. In the experiments, 1000 samples were randomly selected for each working condition, and the ratio of the number of training samples in the source domain and the target domain was 1:1. And 1000

Table 2 Fault classification results of different methods on the bearing datasets of CWRU (accuracy %)

S→T	TCA	Deep CORAL	DDC	DAN	DCTLN	Proposed method
0→1	62.50	98.11	98.24	99.38	99.99	99.74
0→2	65.54	83.35	80.25	90.04	99.99	***100***
0→3	74.49	75.58	74.17	91.48	93.38	***100***
1→0	63.63	90.04	88.96	99.88	99.99	99.71
1→2	64.37	99.25	91.17	99.99	100	***100***
1→3	79.88	87.81	83.70	99.47	100	***100***
2→0	59.05	86.18	67.90	94.11	95.05	***99.23***
2→1	63.39	89.31	90.64	95.26	99.99	99.98
2→3	65.57	98.07	88.28	100	100	***100***
3→0	72.92	76.49	74.60	91.21	89.26	***99.66***
3→1	68.93	79.61	74.77	89.95	86.17	***99.83***
3→2	63.97	90.66	96.70	100	99.98	***100***
Avg	67.02	87.87	84.12	95.9	96.16	***99.85***

The bold and italicized numbers indicate that UDA-BFD-MDD (proposed method) achieved the best transfer learning performance in the corresponding transfer task. The non bold and italicized numbers indicate that other model achieved the best transfer learning performance in the corresponding transfer task samples were randomly selected in the target domain for validation. In order to reduce the effect of random results, the experiments repeated the training 10 times and took the average value. The results are shown in Table 2.

According to the comparative experimental results in Table 2, after using UDA-BFD-MDD for transfer, the average accuracy rate of all transfer tasks is 99.85%, which is higher than that of other classic transfer learning algorithms. Not only that, through analysis of experimental results in Table 2, UDA-BFD-MDD has the following advantages:

1. Not only can a 100% accuracy rate be achieved in tasks that other methods can also achieve, but also when completing tasks such as transfer tasks 0 → 3, 2 → 0, 3 → 0, 3 → 1 and other methods with poor transfer effects, UDA-BFD-MDD can also achieve a higher accuracy rate.
2. The diagnosis results of UDA-BFD-MDD are relatively stable. The accuracy of different transfer tasks fluctuates within 1%, while the fluctuations of other transfer learning algorithms are about 10–20%. The accuracy of UDA-BFD-MDD is greater than 99% in all transfer tasks, while the accuracy of other methods is less than 90% on some difficult transfer tasks.
3. Compared with other methods, the results of UDA-BFD-MDD in the transfer tasks 0 → 1, 1 → 0, 2 → 1 are slightly worse, but when completing tasks 3 → 0, 3 → 1, the accuracy of UDA-BFD-MDD is about 10% higher than other methods. Therefore, the overall accuracy rate of UDA-BFD-MDD is higher.

4.3 Comparative experiments with different sample sizes

In order to verify the fault diagnosis accuracy of UDA-BFD-MDD under different sample numbers, different target domain data volumes were selected for training in this experiment, and the data volumes were taken as 10, 100, 250, 500, 750 and 1000, respectively. Each working condition is trained 10 times under each data amount, and the final result is taken as the average value. The accuracy results comparison under different sample sizes are shown in Fig. 8.

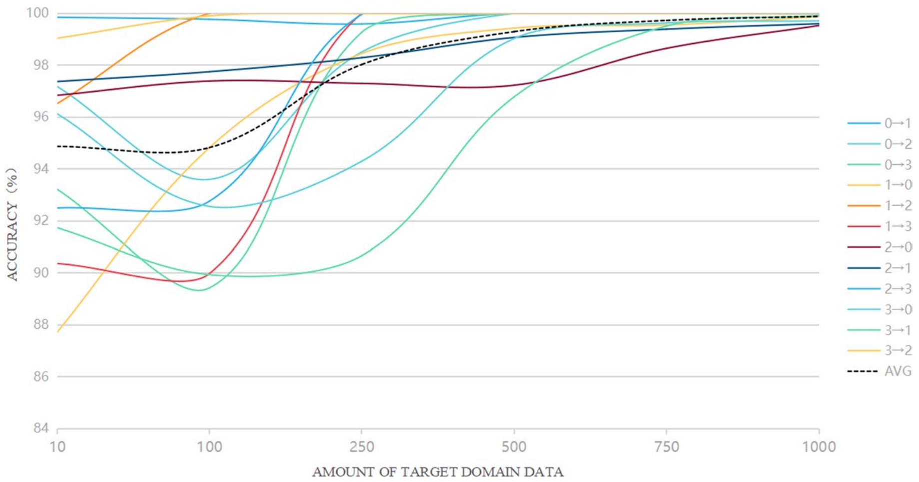


Fig. 8 The accuracy result comparison under different sample sizes

From the results, we found that when the data size is 500, the accuracy rate of all tasks exceeds 96%, and when the data size is 750, the accuracy rate of all tasks exceeds 98%. Under 250 samples, there are only three tasks with 100% accuracy rate. Under 500 samples, there are 7 tasks with 100% accuracy rate. At the same time, we also tried the case of training with only 10 samples and found that the reason for the lower accuracy rate is the large fluctuation of the results in the iterations, and the results in several iterations are the same as the results using 1000 samples. This phenomenon indicates that the learning ability of the proposed algorithm improves with the increase in sample number. It can be seen from the average curve that when the amount of data increases, the fluctuation of the training results during iteration becomes smaller and smaller, and the accuracy of diagnosis also increases. In UDA-BFD-MDD, the target domain data only need half of the source domain data, and its average diagnosis accuracy rate can exceed 99%. There is also an interesting phenomenon in Fig. 8 that some curves, e.g., 1 to 3, decrease first and then increase with the increase in sample volume. The reason behind the phenomenon is that when the number of samples is low, the selection of sample data is random which results in the accuracy fluctuation. With the increase in samples number, the accuracy grows gradually and steadily.

4.4 Data visualization analysis

In order to verify the ability of UDA-BFD-MDD to extract and transfer features, the confusion matrix heatmap and T-SNE are used to analyze the results of the transfer task under different working conditions. The confusion matrix heat map shows the fault classification results before and after the transfer, when the CWRU data set working condition 0, 1, 2, and 3 are, respectively, used as the source domain to transfer to other working conditions. The confusion matrix heatmaps are shown in Figs. 9, 10, 11 and 12, respectively.

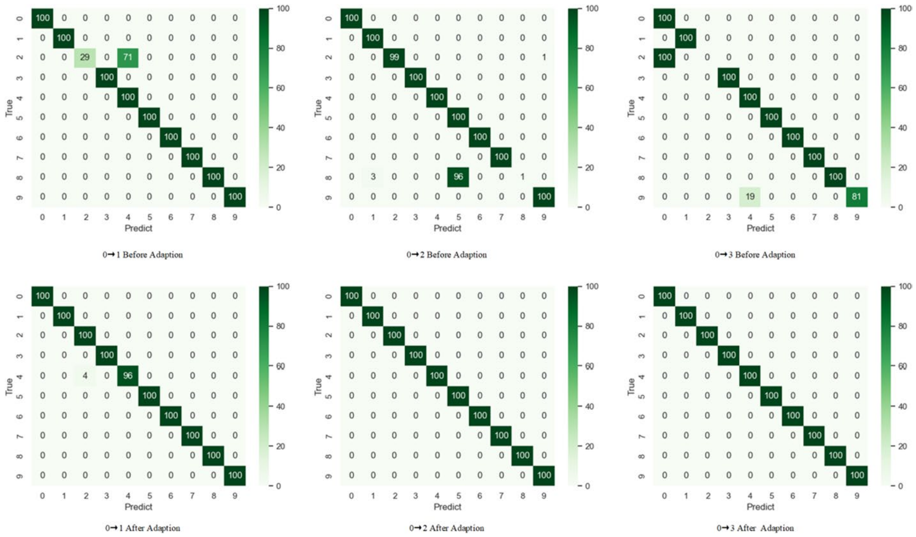
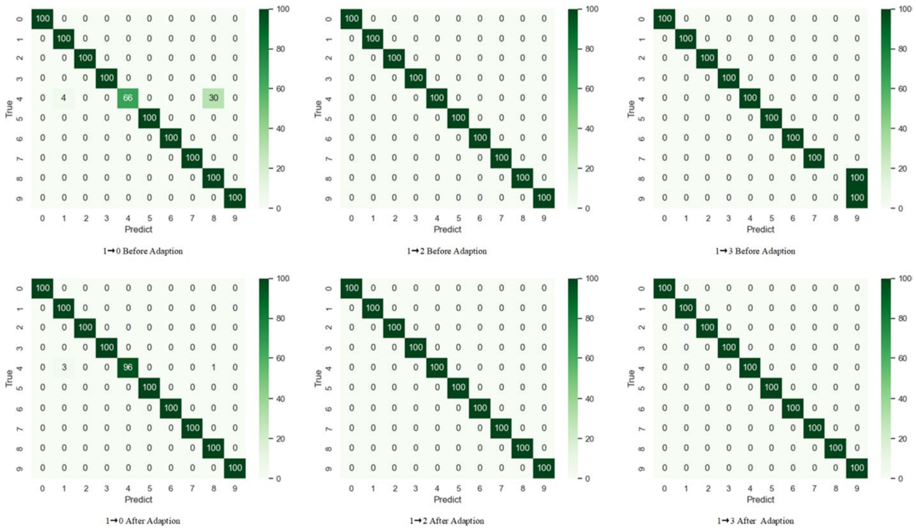
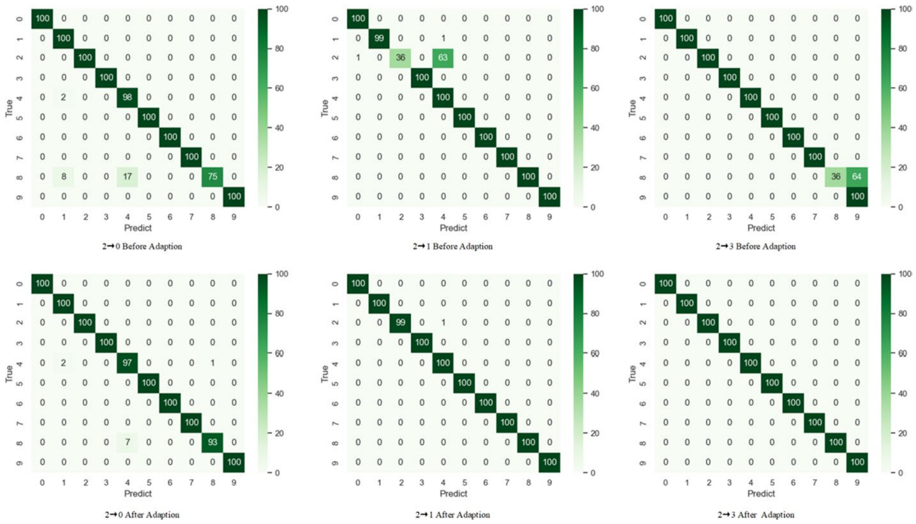


Fig. 9 Confusion matrix heatmap for transfer from case 0 to others



1→0 Before Adaption
1→2 Before Adaption
1→3 Before Adaption
1→0 After Adaption
1→2 After Adaption
1→3 After Adaption

Fig. 10 Confusion matrix heatmap for transfer from case 1 to others



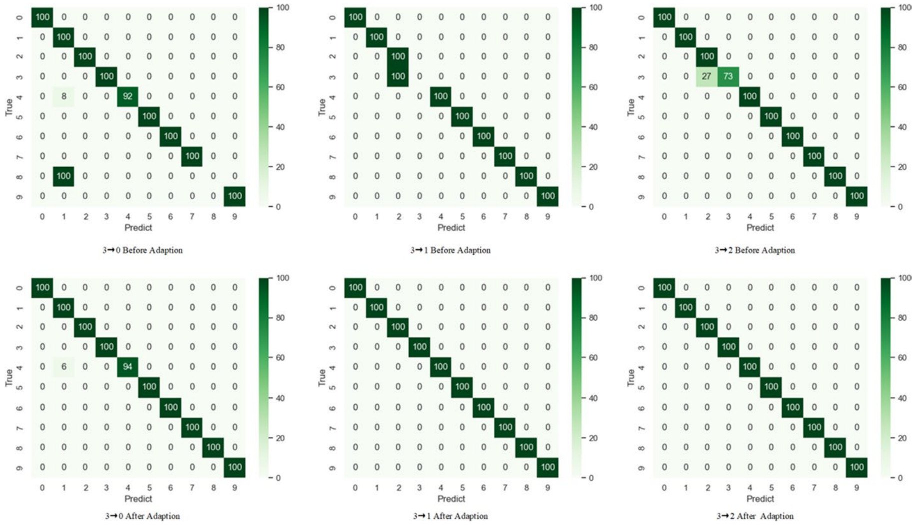
2→0 Before Adaption
2→1 Before Adaption
2→3 Before Adaption
2→0 After Adaption
2→1 After Adaption
2→3 After Adaption

Fig. 11 Confusion matrix heatmap for transfer from case 2 to others

As shown in Fig. 9, in the transfer from working condition 0 to others, the model before the transfer is less effective in classifying fault 2, and the model classifies fault 2 as fault 4 or fault 0. In task 0 → 2, fault 8 is mainly incorrectly classified as fault 5. After the transfer, except for task 0 → 1, the accuracy rate of other tasks reaches 100%.

As illustrated in Fig. 10, in task 1 → 2, the model before transfer can obtain 100% accuracy rate, indicating that the model established in case 1 can be directly transferred to case 2. In tasks 1 → 0 and 1 → 3, the incorrect classification of fault 4 and fault 8 before transferring is solved after transferring.

As depicted in Fig. 11, in task 2 → 0 and task 2 → 3, fault 8 is main misclassification before transferring. After the transfer, the model transferred to working condition 3 has better classification accuracy on fault 8, while the model transferred to working


3→0 Before Adaption

3→1 Before Adaption

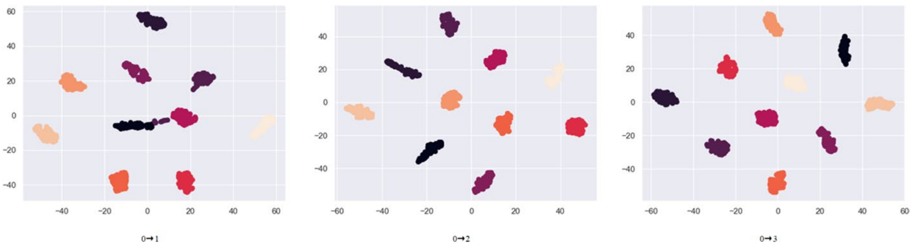
3→2 Before Adaption

3→0 After Adaption

3→1 After Adaption

3→2 After Adaption

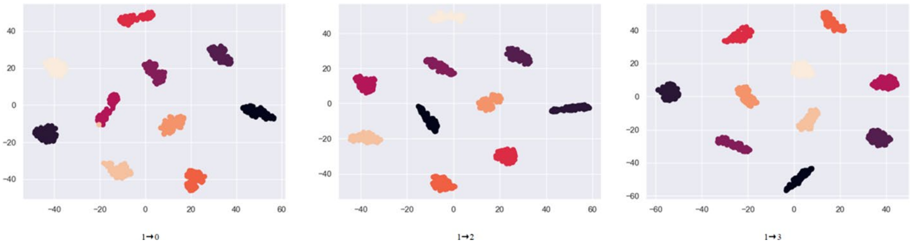
Fig. 12 Confusion matrix heatmap for transfer from case 3 to others


0→1

0→2

0→3

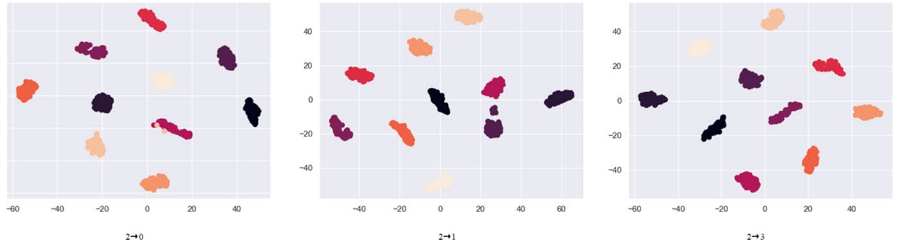
Fig. 13 T-SNE for transfer from case 0 to others


1→0

1→2

1→3

Fig. 14 T-SNE for transfer from case 1 to others


2→0

2→1

2→3

Fig. 15 T-SNE for transfer from case 2 to others

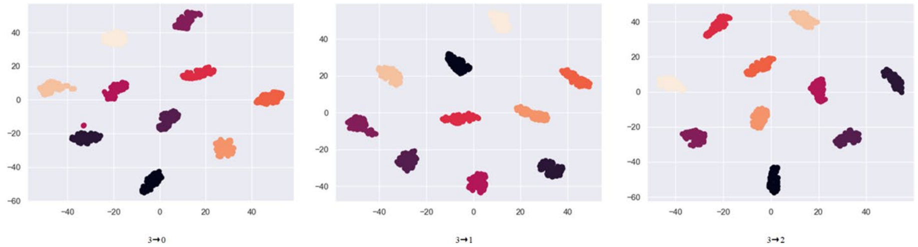


Fig. 16 T-SNE for transfer from case 3 to others

condition 0 has great classification accuracy improvement for fault 8, but no improvement for fault 4.

As shown in Fig. 12, in tasks 3 → 1 and 3 → 2, fault 3 is incorrectly classified as fault 2 before transfer. And in 3 → 0, the model cannot correctly classify fault 8. It is clear that the transfer mainly occurs on faults 4 and 8, indicating that the characteristics of the two faults are similar to other faults in the target domain, and they need to be transferred to correctly be classified. After transferring, except for fault 4, other faults can all reach high classification accuracy.

To intuitively observe the transfer process of UDA-BFD-MDD to the bearing fault features, T-SNE is used to reduce the dimension of the features. Figures 13, 14, 15 and 16 show visualizations of the T-SNE for the transfer tasks under different conditions.

From Figs. 13, 14, 15 and 16, it can be found that after the transfer, in the transfer task with case 0 as the target domain, there are very few points that overlap with other clusters. It shows that the characteristics of some faults in the source domain are similar to that of faults in case 0. In addition, the results of fault classification tend to be more concentrated in the transfer from low-horsepower conditions to high-horsepower conditions, and the distance between classes is farther than the transfer from high-horsepower to low-horsepower conditions. It shows that the transfer effect of high-horsepower features is better than that of low-horsepower features.

5 Conclusion

An unsupervised domain adaptive bearing fault diagnosis method (UDA-BFD-MDD) proposed in this paper uses an adversarial method to align the features of data from source and target domain. Then, the model constructed based on data from source domain can be transferred to target domain data without labels. Therefore, UDA-BFD-MDD avoids the cost of model rebuilding and labeling on target domain data, which enables it to be applied in real industry scenarios. In addition, aiming at the high similarity of some fault features in the bearing fault data, maximum domain discrepancy is exploited, which can find the positive transfer information as much as possible and improve performance of transferring. The experimental results confirm that UDA-BFD-MDD achieves an average accuracy rate of 99.85% for transfer tasks on the CWRU data set, higher than other transfer learning-based methods, and can also perform correctly under small sample data.

Acknowledgements
The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the article.

Author contributions

Each author's contribution to this research work is as follows: CW conceived and designed the experiments, analyzed and interpreted the data and wrote the manuscript. SW contributed to the experimental design, performed the experiments and analyzed the data. XS provided technical support and was expert in data analysis, reviewed and revised the manuscript and analyzed the data.

Funding

This work was supported by the National Natural Science Foundation of China (Nos. 62076215 and 61502411) and the New Generation Information Technology Innovation Project of the Ministry of Education of China (No. 2020ITA02057).

Availability of data and materials

Please contact authors for data requests.

Declaration

Competing interests

The authors declare that they have no competing interests.

Received: 18 September 2023 Accepted: 20 December 2023
Published online: 11 January 2024

References

1. J. Wan, B. Chen, M. Imran, F. Tao, D. Li, C. Liu, S. Ahmad, Toward dynamic resources management for IoT-based manufacturing. IEEE Commun. Mag. **56**(2), 52–59 (2018)
2. B. Wang, F. Tao, X. Fang, C. Liu, Y. Liu, T. Freiheit, Smart manufacturing and intelligent manufacturing: a comparative review. Engineering **7**(6), 738–757 (2021)
3. S. Nandi, H.A. Toliyat, X. Li, Condition monitoring and fault diagnosis of electrical motors: a review. IEEE Trans. Energy Convers. **20**(4), 719–729 (2005)
4. P. Nunes, J. Santos, E. Rocha, Challenges in predictive maintenance: a review. CIRP J. Manuf. Sci. Technol. **40**, 53–67 (2023)
5. H. Wang, W. Zhang, D. Yang, Y. Xiang, Deep-learning-enabled predictive maintenance in industrial internet of things: methods, applications, and challenges. IEEE Syst. J. **17**(2), 2602–2615 (2023)
6. J. Jiang, F. Liu, Y. Liu, Q. Tang, B. Wang, G. Zhong, W. Wang, A dynamic ensemble algorithm for anomaly detection in IoT imbalanced data streams. Comput. Commun. **194**(10), 250–257 (2022)
7. O. Das, D.B. Das, D. Birant, Machine learning for fault analysis in rotating machinery: a comprehensive review. Heliyon **9**(6), e17584 (2023)
8. J. Jiang, F. Liu, W.W.Y. Ng, Q. Tang, W. Wang, Q.-V. Pham, Dynamic incremental ensemble fuzzy classifier for data streams in green internet of things. IEEE Trans. Green Commun. Netw. **6**(3), 1316–1329 (2022)
9. Y. Ren, Y. Leng, J. Qi, P.K. Sharma, J. Wang, Z. Almakhadmeh, A. Tolba, Multiple cloud storage mechanism based on blockchain in smart homes. Future Gener. Comput. Syst. **115**(2), 304–313 (2021)
10. D.-T. Hoang, H.-J. Kang, A survey on deep learning based bearing fault diagnosis. Neurocomputing **335**(2019), 327–335 (2019)
11. Z. Zhu, Y. Lei, G. Qi, Y. Chai, N. Mazur, Y. An, X. Huang, A review of the application of deep learning in intelligent fault diagnosis of rotating machinery. Measurement **206**, 112346 (2023)
12. L. Zhang, J. Wang, W. Wang, Z. Jin, Su. Yansen, H. Chen, Smart contract vulnerability detection combined with multi-objective detection. Comput. Netw. **217**(9), 1–13 (2022)
13. M. Hakim, A.A. Borhana Omran, A.N. Ahmed, M. Al-Waily, A. Abdellatif, A systematic review of rolling bearing fault diagnoses based on deep learning and transfer learning: taxonomy, overview, application, open challenges, weaknesses and recommendations. Ain Shams Eng. J. **14**(4), 101945 (2023)
14. S.J. Pan, Q. Yang, A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
15. B. Yang, Y. Lei, F. Jia, S. Xing, An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings. Mech. Syst. Signal Process. **122**, 692–706 (2019)
16. X. Li, W. Zhang, Q. Ding, J.-Q. Sun, Multi-layer domain adaptation method for rolling bearing fault diagnosis. Signal Process. **157**, 180–197 (2019)
17. B.R. Yang, Q. Li, L. Chen, C.Q. Shen, Bearing fault diagnosis based on multilayer domain adaptation. Shock. Vib. **2020**(1–2), 1–11 (2020)
18. Z.H. Liu, B.L. Lu, H.L. Wei, L. Chen, X.H. Li, M. Rätsch, Deep adversarial domain adaptation model for bearing fault diagnosis. IEEE Trans. Syst. Man Cybern. Syst. **51**(7), 4217–4226 (2021)
19. X. Wang, F. Liu, Triplet loss guided adversarial domain adaptation for bearing fault diagnosis. Sensors **20**(1), 320 (2020)
20. V.M. Patel, R. Gopalan, R. Li, R. Chellappa, Visual domain adaptation: a survey of recent advances. IEEE Signal Process. Mag. **32**(3), 53–69 (2015)
21. S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis. IEEE Trans. Neural Netw. **22**(2), 199–210 (2011)
22. Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**(59), 1–35 (2016)
23. H. Zhao, R.T.D. Combes, K. Zhang, G.J. Gordon, On learning invariant representations for domain adaptation, in: *Proceedings of International Conference on Machine Learning (PMLR 2019)*, pp. 7523–7532 (2019).
24. A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, A. Smola, A kernel two-sample test. J. Mach. Learn. Res. **13**(3), 723–773 (2012)

25. M. Long, Y. Cao, J. Wang, M.I. Jordan, Learning transferable features with deep adaptation networks, in: *Proceedings of International Conference on Machine Learning (PMLR, 2015)*, pp. 97–105 (2015).
26. L. Guo, Y. Lei, S. Xing, T. Yan, N. Li, Deep convolutional transfer learning network: a new method for intelligent fault diagnosis of machines with unlabeled data. IEEE Trans. Ind. Electron. **66**(9), 7316–7325 (2018)
27. B. Sun, K. Saenko, Deep coral: correlation alignment for deep domain adaptation, in: *Proceedings of European Conference on Computer Vision*, (2016), pp. 443–450.
28. I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: *Proceedings of the 27th International Conference on Neural Information Processing (NIPS 2014)*, (2014), pp. 2672–2680.
29. H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, Domain-adversarial neural networks. arXiv preprint arXiv:1412.4446 (2014).
30. E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: *The Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 2962–2971 2017.
31. E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, T. Darrell, Deep domain confusion: maximizing for domain invariance. arXiv 2014, arXiv:1412.3474.
32. G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv preprint arXiv:1207.0580, 2012.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.