IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 31, 2020, accepted January 17, 2021, date of publication January 20, 2021, date of current version January 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053075

Deep Transfer Network With Multi-Kernel Dynamic Distribution Adaptation for Cross-Machine Fault Diagnosis

MINGZHU LV[1,2], SHIXUN LIU[1,3], XIAOMING SU[1], AND CHANGZHENG CHEN[1]
[1]School of Mechanical Engineering, Shenyang University of Technology, Shenyang 110870, China
[2]School of Automatic Control Engineering, Liaoning Equipment Manufacture College of Vocational Technology, Shenyang 110161, China
[3]CQC (ShenYang) North Laboratory, Shenyang 110164, China

Corresponding author: Mingzhu Lv (zhaogx@sut.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 51675350.

ABSTRACT Recently, various deep learning models, which are mainly based on data-driven algorithms, have received more and more attention in the field of intelligent fault diagnosis and prognostics. However, there are two major assumptions accepted by default in the existing studies: 1) The training (source domain) and testing (target domain) data sets obey the same feature distribution; 2) Sufficient labeled data with fault information is available for model training. In real industrial scenarios, especially for different machines, these assumptions are mostly invalid, which makes it a huge challenge to build reliable diagnostic model. Motivated by transfer learning, we present a novel intelligent method named deep transfer network (DTN) with multi-kernel dynamic distribution adaptation (MDDA) to address the problem of cross-machine fault diagnosis. In the proposed approach, the DTN has wide first-layer convolutional kernel and several small convolutional layers, which is utilized to extract transferable features across different machines and suppress high frequency noise. Then, the MDDA method constructs a weighted mixed kernel function to map different transferable features to a unified feature space, and the relative importance of the marginal and conditional distributions are also evaluated dynamically. The proposed method is verified by three transfer learning tasks of bearings, in which the health states of wind turbine bearings in real scenario are identified by using diagnosis knowledge from two different bearings in laboratories. The results show that the proposed method can achieve higher diagnosis accuracy and better transfer performance even under different noisy environment conditions than many other state-of-the-art methods. The presented framework offers a promising approach for cross-machine fault diagnosis.

INDEX TERMS Deep transfer network, multi-kernel dynamic distribution adaptation, cross-machine fault diagnosis, transfer learning, bearings.

I. INTRODUCTION

Rolling element bearings are key components of the rotating machinery, whose health states directly affect the performance, stability and service life of the machinery. According to statistics, in all rotating machine faults, bearing failure accounts for about 30% [1], [2]. Therefore, it is of great importance to accurately diagnose and identify the health status of the bearings. In recent years, diverse data-driven intelligent fault diagnosis methods have received more and more attention due to their superior performances, such as low cost, high precision and fast response. With the continuous development of deep learning, intelligent fault diagnosis of bearings has made marvelous achievements [3]-[5]. However, the main disadvantage of most existing methods is that they are restricted by two assumptions. One is that the training and testing data sets are taken from the same feature distribution; the other is that there are sufficient labeled data with fault information for model training. In real industrial scenarios, on the one hand, due to different operating conditions, such as varying loads and operating speeds, the domain shift phenomenon of training and testing data sets is widespread, which can greatly deteriorate the generalization ability of the traditional machine learning methods. On the other hand,

The associate editor coordinating the review of this manuscript and approving it for publication was Xiao-Sheng Si.

since it takes a long time for a machine to go from run to failure, the fault data itself is difficult to obtain, not to mention the labeled fault data. It is a time-consuming and expensive task.

Transfer learning is a promising method to tackle the aforementioned problems, and has proven its wide applicability spanning through various fields [6]–[8]. In transfer learning, the training data sets and the testing data sets are defined as source domain and target domain respectively. Compared with the source domain data, the target domain data has relevant knowledge but different distribution. Different from traditional machine learning methods, the goal of transfer learning is to enhance the performance of the model and reduce the quantity of required sample in the target domain using transferable features or diagnosis knowledge from source domain [9]. In order to achieve this goal, the feature-based approach has been widely studied as one of the common used transfer learning methods [10]. This method focuses on learning a feature mapping, which extracts transferable features from source domain and target domain to reduce cross-domain distribution discrepancy. With the introduction of deep learning, deep structure models are used to automatically extract transferable features from different domains. Some scholars have begun to engage in some studies by the feature-based transfer learning approach. Yang *et al.* [11] used Convolution Neural Network (CNN) to extract the transferable features of the raw vibration data from the source and target domains, and then the regularization terms are introduced, which are employed to impose constraints on the parameters of CNN in order to reduce the distribution divergence between domains and the among-class distance of the learned features. Wang *et al.* [12] first adopted the improved ResNet50 to extract low-level features, then constructed a multi-scale feature learner to analyze these features, and took the obtained high-level features as the input of the classifier. Chatterjee and Dethlefs [13] utilized an exponential linear activation function to improve the quality of mapped vibration data, and adopted non-negative constraints to modify the loss function so as to improve the effect of feature-based transfer learning. However, these methods only minimize the distance between cross-domain feature distributions, and do not realize the distribution alignment. Thus, feature distribution alignment is still a challenge for domain adaptation. Most of the existing methods try to align the marginal distribution [14], [15], or the conditional distribution [16], or assume that both distributions are equally important [17]. In the field of computer vision, the latest research has shown that perform dynamic distribution adaptation (DDA) can obtain better transfer performance [18]. Wang *et al.* [19] first proposed Dynamic Distribution Adaptation Network (DDAN) to use the deep neural network in learning end-to-end transfer classifier. Although DDAN has achieved good results in image recognition, as for bearing fault diagnosis, some cross-characteristics of different domains need to be obtained in advance. Moreover, the DDAN model has poor anti-noise ability.

On the basis of absorbing and drawing upon informed research, this paper presented a novel intelligent fault diagnosis framework, named deep transfer network (DTN) with multi-kernel dynamic distribution adaptation (MDDA) for cross-machine fault diagnosis. The contributions of this paper are summarized as follows.

1) We present a MDDA method by introducing mixed kernel functions. The proposed method can extract richer features from cross-domain data without feature transformation. By adjusting the balance factor of the kernel function, different mapping effects of different features are realized.
2) A novel DTN is developed to work directly on raw vibration signals. Moreover, this model perform well under noisy environment conditions with no pre-denoising methods.
3) The DTN with MDDA method can utilize the diagnosis knowledge of labeled source domain data to realize the prediction of unlabeled target domain data. Three different transfer scenarios from different machines were used to verify the effectiveness of the proposed method. Compared with other state-of-the-art methods, the presented framework obtains higher classification accuracy and superior transfer performance.

The remainder of this paper is organized as below. In Section 2, we describe the transfer learning tasks and introduce the idea of the maximum mean discrepancy (MMD). In Section 3, the proposed intelligent fault diagnosis framework are explained including the architecture of the proposed network, fully-connected layers domain adaptation and training procedure. In Section 4, we conduct three transfer learning tasks, and corresponding results are listed. In Section 5, the analysis and discussion about the results of the experiments are given. Finally, the conclusions are drawn in Section 6.

II. PRELIMINARIES

A. PROBLEM DESCRIPTION

Let D_s and D_t represent the source and target domains respectively. The sample spaces are denoted as $X_s \in D_s$ and $X_t \in D_t$, then the data samples drawn from the source domain and target domain are $x_s \in X_s$ and $x_t \in X_t$ respectively. We also define that the label space $Y = \{1, 2, \cdots, C\}$, which contains C kinds of health states. Here, we assume that the source and target domains have the same health states categories.

In this paper, we are dedicated to studying the tasks of intelligent fault diagnosis between different machines. Assume that the data samples from the source domain and the target domain are subject to the marginal probability distribution $P(X_s)$ and $P(X_t)$, as well as conditional

probability distribution $Q(Y_s|X_s)$ and $Q(Y_t|X_t)$, $P(X_s) \neq P(X_t)$, $Q(Y_s|X_s) \neq Q(Y_t|X_t)$. Therefore, the transfer learning from the source domain to the target domain has the following definition.

1) The source domain contains n_s labeled samples, ie., $X_s = \{x_i^s, y_i^s\}_{i=1}^{n_s}$.

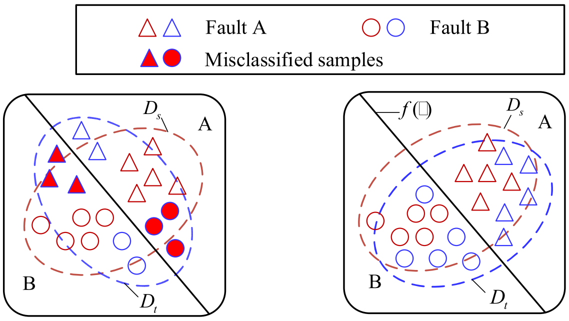
2) The target domain contains n_t unlabeled samples, ie., $X_t = \left\{x_j^t\right\}_{j=1}^{n_t}$.
3) The goal of domain adaptation is to learn a feature mapping and minimize the discrepancy between marginal distribution and conditional distribution, i.e.,

$$\min D(P_s(\phi(X_s)), P_t(\phi(X_t))) \tag{1}$$

$$\text{and } \min D(Q_s(Y_s|\,\phi(X_s)), Q_t(Y_t|\,\phi(X_t))) \tag{2}$$

where D is the function to evaluate the domain discrepancy, $\phi(\cdot)$ is the mapping function.

In practical applications, the vibration data in the source domain and the target domain are collected from different bearings. Thus, the distribution discrepancy of these data are serious. If the fault identification is imposed directly on these data through the domain-shared classifier, it will generate very poor diagnosis results. As shown in Fig. 1(a), a domain-shared classifier $f(\cdot)$ has been just trained with source domain samples using the structural risk minimization theory [20], which means that it can complete the classification task based on the learned features. This classifier is also expected to be well applied in the target domain. However, in the target domain, we got very unsatisfactory classification results, whose generalization error is enlarged. In other words, the reason for the low classification accuracy of the classifier $f(\cdot)$ is the serious distribution discrepancy between the learned features from the source domain and the target domain. Therefore, in order to improve the classification accuracy, we need build an intelligent diagnosis model that can extract transferable features to reduce cross-domain distribution discrepancy. Thus, the domain-shared classifier $f(\cdot)$ can also minimize the structural risk on the target domain. From Fig. 1(b), the intelligent diagnosis model is expected to learn transferable features with similar distributions. Finally, the domain-shared classifier $f(\cdot)$ can correctly distinguish the target domain samples using the diagnosis knowledge provided by the source domain.



(a) (b)

FIGURE 1. The identification results of intelligent diagnosis model: (a) without domain adaptation, and (b) with domain adaptation.

B. MAXIMUM MEAN DISCREPANCY

For two domains with independent and different distributions, the distance between domains is usually adapted to measure the distribution divergence. Maximum mean discrepancy (MMD) is widely used in domain adaptation, which is a non-parametric measurement method [21]. If the data sets obey the probability distribution p and q respectively, the MMD between the two data sets can be defined as follows.

$$D_{\mathcal{H}}(X_s, X_t) := \sup_{\phi\in\mathcal{H}} \left\{E_{X_s\sim p}[\phi(X_s)] - E_{X_t\sim q}[\phi(X_t)]\right\} \tag{3}$$

where D is the distance to evaluate domain deviation, $\mathcal{H}$ is the reproduced kernel Hilbert space (RKHS), sup$(\cdot)$ is the supremum of the input aggregate, $\phi(\cdot)$ represents the nonlinear mapping function from $X \rightarrow \mathcal{H}$, $E(\cdot)$ denotes the mean of the embedded samples.

In statistics, it is called integral probability metric. To calculate this discrepancy, a biased empirical estimate of MMD is as follows:

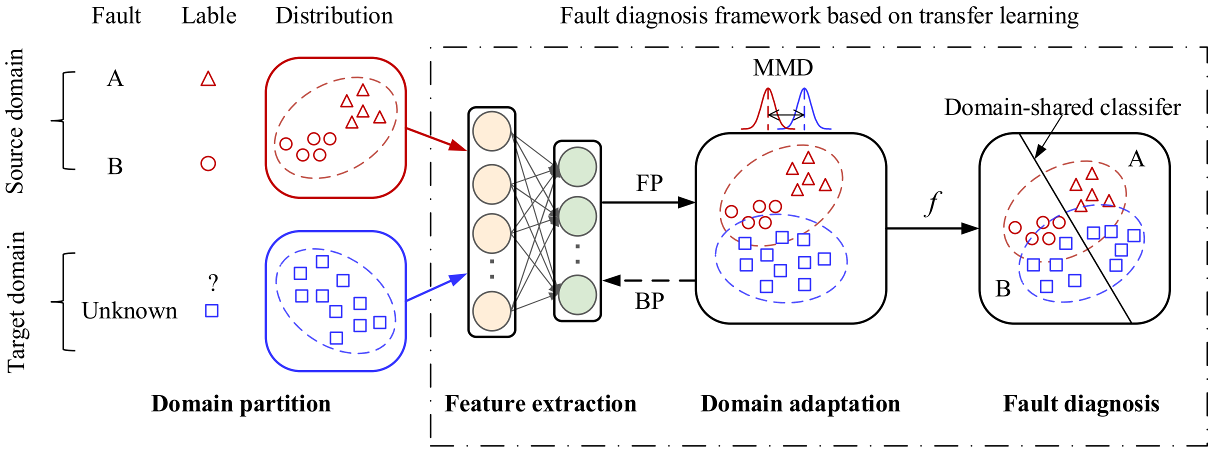
$$\hat{D}_{\mathcal{H}}(X_s, X_t) = \left\| \frac{1}{n_s}\sum_{i=1}^{n_s}\phi(x_i^s) - \frac{1}{n_t}\sum_{j=1}^{n_t}\phi(x_j^t) \right\|_{\mathcal{H}}^2 \tag{4}$$

It can be seen from (4) that the empirical estimation of the deviation between the two distributions can be considered as the distance between the mean of the two data sets in RKHS. When MMD is close to 0, it means that the two distributions are aligned. In transfer learning, MMD is usually used to construct constraints on feature learning to make the distribution in different domains more similar.

III. THE PROPOSED APPROACH

A. OVERVIEW OF THE METHODOLOGY

In this paper, our idea is to establish an intelligent diagnosis model inspired by transferable feature methods. Generally speaking, the proposed method consists of four stages: domain partition, feature extraction, domain adaptation, and fault diagnosis, as shown in Fig. 2. In the stage of domain partition, the diagnosis knowledge is provided by source domain data, while unlabeled target domain data is expected to be correctly identified by using knowledge transfer methods. As for feature extraction, the transferable features, which are simultaneously extracted by the same nonlinear feature mapping from samples in the source and target domains. For domain adaptation, the MMD algorithm is utilized to measure the distribution divergence of learned features. After that, a new method named multi-kernel dynamic distribution adaptation (MDDA) constructs a weighted mixed kernel function to map the transferable features to a unified high-dimensional feature space, and dynamically evaluates the relative importance of marginal probability distribution (MPD) and conditional probability distribution (CPD), so as to minimize the discrepancy between the two distributions, and finally obtain a target classifier by the structural risk minimization (SRM) principle. For fault diagnosis, by using the domain-shared classifier, the unlabeled target domain samples can be correctly classified.



classifier by the structural risk minimization (SRM)

FIGURE 2. Overview of the methodology.

B. BASIC THEORY OF THE METHODOGY

1) DYNAMIC DISTRIBUTION ADAPTATION

According to (1) and (2), the objective of domain adaptation is to minimize the MPD and the CPD of the two data sets in RKHS. On one hand, we can apply MMD to handle the MPD. The formula is described as

$$MMD^2_{\mathcal{H}}(P_s, P_t) = \left\| \frac{1}{n_s}\sum_{i=1}^{n_s}\phi(x_i^s) - \frac{1}{n_t}\sum_{j=1}^{n_t}\phi(x_j^t) \right\|^2_{\mathcal{H}} \tag{5}$$

On the other hand, the conditional distribution in (2) is intractable in the absence of classification labels. According to Bayesian rule [22], we rewrite formula (2) into the following form:

$$\min D(\frac{Q_s(\phi(X_s)|\,Y_s)\cdot P_s(Y_s)}{P_s(\phi(X_s))}, \frac{Q_t(\phi(X_t)|\,Y_t))\cdot P_t(Y_t)}{P_t(\phi(X_t))}) \tag{6}$$

In the paper, we have the assumption of $P_s(Y_s) = P_t(Y_t)$. In other words, suppose that the labels of the source and the target domains have the similar distributions. If the marginal distribution satisfies the (1), the conditional distribution problem becomes

$$\min D(Q_s(\phi(X_s)|\,Y_s), Q_t(\phi(X_t)|\,Y_t)) \tag{7}$$

The objective function of (7) is noted as CDA. In domain adaptation, this step is essential. However, it still can not handle as Y_t is unknown. To address the problem, we can use the labeled source domain samples to train a simple classifier to obtain pseudo-labels on the target domain, then the interclass MMD distance is expressed as

$$\begin{aligned} &MMD^2_{\mathcal{H}}(Q_s^{(c)}, Q_t^{(c)}) \\ &= \left\| \frac{1}{n_s^{(c)}}\sum_{x_i^s\in D_s^{(c)}}\phi(x_i^s) - \frac{1}{n_t^{(c)}}\sum_{x_j^t\in D_t^{(c)}}\phi(x_j^t) \right\|^2_{\mathcal{H}} \end{aligned} \tag{8}$$

where $c \in \{1, 2, \cdots, C\}$ is the c-th category, $D_s^{(c)} = \{x_i : x_i \in D_s \cap y(x_i) = c\}$, $y(x_i)$ is the true label, and $n_s^{(c)}$ is the total number of all label c for the samples in the source domain. $D_t^{(c)} = \{x_j : x_j \in D_t \cap \hat{y}(x_j) = c\}$, $\hat{y}(x_j)$ is the pseudo label, and $n_t^{(c)}$ is the total number of all pseudo label c for the samples in the target domain.

It is certain that, although there are probably large deviations in the initial pseudo labels, we can iteratively update the pseudo labels in the process of model training, the correctness of the pseudo labels can be improved to ensure that the divergence in conditional distribution becomes smaller and smaller. Through (5) and (8), the marginal (P) and conditional (Q) distributions can be aligned, but the two distributions are not equally important in real-world applications. For instance, when there is a large difference between data sets, the discrepancy between P_s and P_t is more dominant. In contrast, when the data sets are similar, the distribution divergence in each class (Q_s and Q_t) is more dominant. Therefore, an adaptive factor is introduced to dynamically adjust the importance of these two distributions, dynamic distribution adaptation (DDA) can be written as

$$\begin{aligned} &D_{\mathcal{H}}(D_s, D_t) \\ &= (1-\mu)MMD^2_{\mathcal{H}}(P_s, P_t) + \mu\sum_{c=1}^{C}MMD^2_{\mathcal{H}}(Q_s^{(c)}, Q_t^{(c)}) \end{aligned} \tag{9}$$

where $\mu \in [0, 1]$ is the adaptive factor.

From (9), when $\mu \to 0$, it shows that the feature distribution of the source domain and the target domain are very different, therefore, the adaptation of the MPD is relatively important. When $\mu \to 1$, it shows that the discrepancy in feature distributions is small, and the sparsity between each class needs to be adjusted. Therefore, the alignment of the CPD is more important. When $\mu = 0.5$, the two distributions are treated equally, as in existing methods [23]. By learning the optimal adaptation factor μ_{opt}, DDA can be used to address different domain adaptation problems. Putting (5) and (8) into (9), the following formula can be obtained.

$$\begin{aligned} D_{\mathcal{H}}(D_s, D_t) &= (1-\mu)\left\| \frac{1}{n_s}\sum_{i=1}^{n_s}\phi(x_i^s) - \frac{1}{n_t}\sum_{j=1}^{n_t}\phi(x_j^t)\right\|_{\mathcal{H}}^2 \\ &+\mu\sum_{c=1}^{C}\left\| \frac{1}{n_s^{(c)}}\sum_{x_i^s\in D_s^{(c)}}\phi(x_i^s) - \frac{1}{n_t^{(c)}}\sum_{x_j^t\in D_t^{(c)}}\phi(x_j^t)\right\|_{\mathcal{H}}^2 \end{aligned} \quad (10)$$

By taking advantage of representer theorem and matrix tricks [24], (10) can be written as

$$D_{\mathcal{H}}(D_s, D_t) = \mathrm{tr}(KM) \quad (11)$$

where $\mathrm{K} = \phi([X_s, X_t])^{\mathrm{T}}\phi([X_s, X_t]) \in R^{(n_s+n_t)\times(n_s+n_t)}$ is the kernel matrix with $k_{ij} = k(x_i, x_j)$ in RKHS, $\mathrm{tr}(\cdot)$ denotes the trace operation, $\mathrm{M} = (1-\mu)\mathrm{M}_0 + \mu\sum_{c=1}^{C}\mathrm{M}_c$ is the MMD matrix with its element calculated by

$$(\mathrm{M}_0)_{ij} = \begin{cases} \frac{1}{n_s^2}, & x_i, x_j \in D_s \\ \frac{1}{n_t^2}, & x_i, x_j \in D_t \\ -\frac{1}{n_s n_t}, & \text{otherwise} \end{cases} \quad (12)$$

$$(\mathrm{M}_c)_{ij} = \begin{cases} \frac{1}{(n_s^{(c)})^2}, & x_i, x_j \in D_s^{(c)} \\ \frac{1}{(n_t^{(c)})^2}, & x_i, x_j \in D_t^{(c)} \\ -\frac{1}{n_s^{(c)}n_t^{(c)}}, & \begin{cases} x_i \in D_s^{(c)}, & x_j \in D_t^{(c)} \\ x_i \in D_t^{(c)}, & x_j \in D_s^{(c)} \end{cases} \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

The above non-convex optimization problem can be transformed into a trace optimization problem by the Lagrange multiplier method.

2) MULTI-KERNEL DYNAMIC DISTRIBUTION ADAPTATION

DDA has two problems that need to be solved: (1) how to obtain the adaptive factor; (2) The convergence efficiency and effect of the MMD criterion depend heavily on the choice of kernel function. However, for a specific application, the optimal kernel function cannot be determined in advance. In order to deal with these problems, this paper presents multi-kernel dynamic distributed adaptation (MDDA) method. By adding a new parameter β, multiple kernel functions are given different weights, so as to better combine the advantages of different kernel functions, which can extract richer features without feature transformation. The formula of the weighted mixed kernel functions is defined as follows:

$$\begin{aligned} k(x_i, x_j) &= (1-\beta)k_{Rbf}(x_i, x_j) + \beta k_{\mathrm{Ploy}}(x_i, x_j) \\ &= (1-\beta)\exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}) + \beta[x_i^{\mathrm{T}}x_j + 1]^d \end{aligned} \quad (14)$$

where k_{Rbf} and k_{Ploy} represent Gaussian radial basis kernel function and polynomial kernel function respectively, and $\beta \in [0, 1]$ controls the weight of the two kernel functions, which is called the balance factor.

MDDA uses $\mathcal{A}$-distance to estimate empirically β. $\mathcal{A}$-distance is defined as the loss of constructing a linear classifier for a binary classification problem [25]. Let $e(h)$ be the error of the linear classifier h distinguishing different domains D_s and D_t, then $\mathcal{A}$-distance can be defined as

$$\mathcal{A}(D_s, D_t) = 2(1 - 2e(h)) \quad (15)$$

Empirically estimate the balance factor β, the formula is as follows:

$$\hat{\beta} = \frac{\mathcal{A}_{\mathrm{Ploy}}(D_s, D_t)}{\mathcal{A}_{\mathrm{Rbf}}(D_s, D_t) + \mathcal{A}_{\mathrm{Ploy}}(D_s, D_t)} \quad (16)$$

where $\mathcal{A}_{\mathrm{Rbf}}(D_s, D_t)$ and $\mathcal{A}_{\mathrm{Ploy}}(D_s, D_t)$ represent the corresponding A-distance after kernel mapping respectively. The larger the value is, the greater difference between the source and the target domains after the kernel mapping, thus, the weight of the corresponding kernel function is smaller, and vice versa.

Similarly, using $\mathcal{A}$-distance to estimate the adaptive factor μ, the formula can be described as follows:

$$\hat{\mu} = \frac{\sum_{c=1}^{C}\mathcal{A}_c(D_s, D_t)}{\mathcal{A}_M(D_s, D_t) + \sum_{c=1}^{C}\mathcal{A}_c(D_s, D_t)} \quad (17)$$

where $\mathcal{A}_c(D_s, D_t)$ represents the CPD for class c, $\sum_{c=1}^{C}\mathcal{A}_c(D_s, D_t)$ represents the $\mathcal{A}$-distance of the CPD for all categories, $\mathcal{A}_M(D_s, D_t)$ represents the $\mathcal{A}$-distance of the MPD for source and target domains.

C. DEEP TRANSFER NETWORK WITH MDDA

Having introduced the basic theory of MDDA, we now turn to establish DTN so as to address the domain adaptation problem under deep learning framework. CNN is a network structure widely used in the field of fault diagnosis. It has excellent characteristics of local connection and weight sharing, including convolutional layers, pooling layers and fully-connected layers [26]. Nevertheless, CNN's network depth, convolution kernel width and appropriate domain adaptation methods etc, which greatly affect the classification accuracy and computational complexity of transfer learning. If the network structure is too shallow, for example, the model used in [27] has a two-layer convolution structure and can only learn low-level simple features. If the network structure is too deep, a large amount of training data is required, and over-fitting is prone to occur. Besides, when 1-D vibration signals are fed into the network, the small convolution kernels (usually 3 × 1) at the first layer are easily interfered by industrial environment noise, most of which belongs to high frequency noise. Therefore, to capture more useful information of vibration signals in the intermediate and low frequency

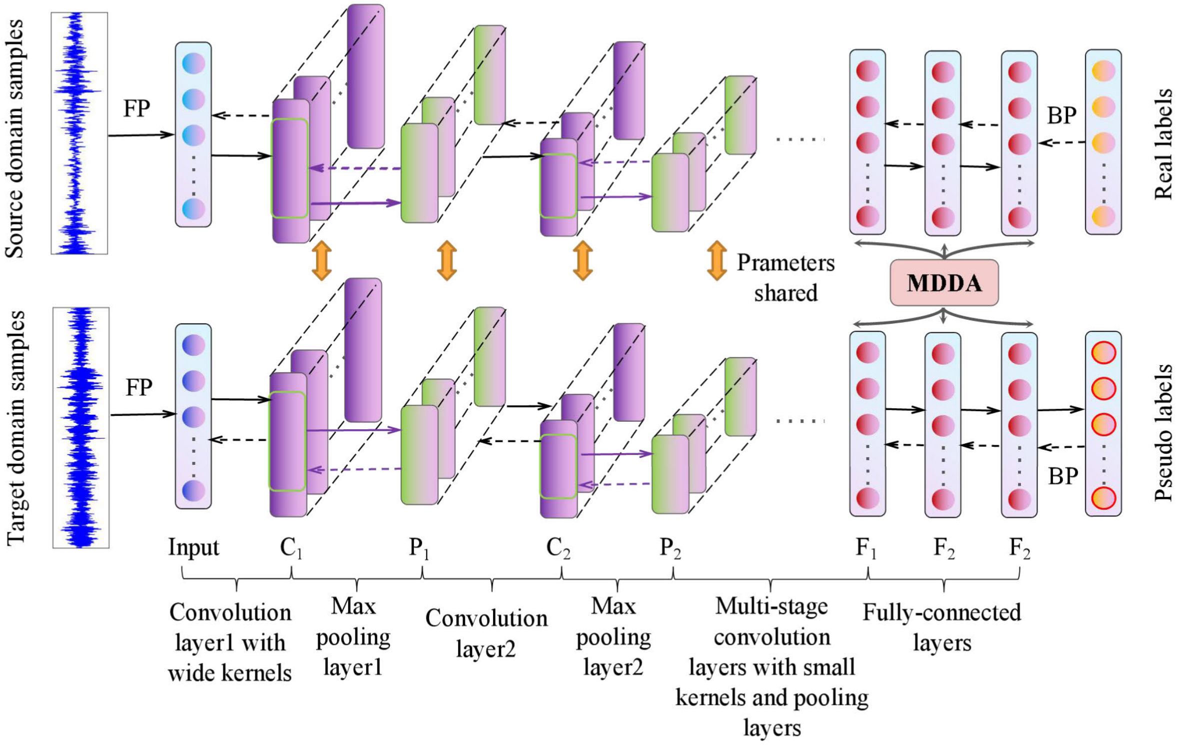


FIGURE 3. Architecture of the proposed DTN with MDDA model.

bands, the wide kernels should be used to extract features, and then successive small kernels, which are utilized to obtain better feature representation. In addition, we embed the MDDA method into the fully-connected layers to align the feature distributions of the source and target domains.

1) ARCHITECTURE OF THE PROPOSED NETWORK

The network structure of the proposed DTN is shown in Fig. 3. According to Ref[10], We use a deep CNN model to extract transferable features from the source and target domains. The raw temporal signals without any artificial processing are used as the input of the first convolutional layer. Unlike the conventional CNN, the first convolutional kernels are wide, while the subsequent convolution kernels are small. By widening the receptive field of the first convolutional layer, the model can extract more features and suppress high-frequency noise. The multi-layer small convolution kernels can build a deeper model, which helps to improve network performance and obtain better feature representation, the parameter set is shown in Table 1. The proposed DTN ties the shared parameter set when the samples both in source and target domains are processed simultaneously. In the convolutional layer, the filter kernel convolves the input local regions, and then generates output features by the activation function. Each filter employs the same kernel which is usually referred to as weight-sharing in the literature.

We adopt k_i^l and b_i^l to denote the weight and bias of the i-th filter kernel in layer l respectively, and use $x_i^{l,D}$ to represent the output feature in layer l, then the output features of the layer $l+1$ can be obtained as

$$x_i^{l+1,D} = \sigma_r(x_i^{l,D} * k_i^l + b_i^l) \tag{18}$$

where $D = \{s, t\}$ denotes the indexes of the source and the target domains, $x_i^{l+1,D} = \left\{x_i^{l+1,s}, x_i^{l+1,t}\right\}$ is the transferable features learned from the features $x_i^{l,D} = \left\{x_i^{l,s}, x_i^{l,t}\right\}$ of the previous layer l. $*$ represents the convolution operation, $\sigma_r(\cdot)$ represents the activation function, generally, the rectified linear unit (ReLU) is employed to accelerate the convergence of the network.

We add a pooling layer after each convolutional layer. It functions as a down-sampling operation, which can reduce the number of the trained parameters and avoid over-fitting. In this study, we use the max pooling form, which can preserve the edges and textures of the transferable features as much as possible, and obtain local- invariant features. The feature after the max-pooling transformation is described as follows:

$$p_i^{l+1,D}(j) = \max_{(j-1)h+1 \le t \le jh} \left\{x_i^{l,D}(t)\right\} \tag{19}$$

where $x_i^{l,D}(t)$ denotes the value of t-th neuron in the i-th frame of layer l, $t \in [(j-1)\omega + 1, j\omega]$, ω is the width of the pooling region, and $p_i^{l+1,D}(j)$ represents the corresponding value of the neuron in layer $l+1$ of the pooling operation.

TABLE 1. Parameter set of proposed DTN model.

Layers	Tied parameters	Activation function	Output size
Input	/	/	1200×1
Convolution1 (C1)	Kernel Size: 64×1, Stride: 16×1, Number: 16	ReLU	128×16
Pooling1(P1)	Kernel Size: 2×1, Stride: 2×1, Number: 16	/	64×16
Convolution2 (C2)	Kernel Size: 3×1, Stride: 1×1, Number: 32	ReLU	64×32
Pooling2(P2)	Kernel Size: 2×1, Stride: 2×1, Number: 32	/	32×32
Convolution3 (C3)	Kernel Size: 3×1, Stride: 1×1, Number: 64	ReLU	32×64
Pooling3(P3)	Kernel Size: 2×1, Stride: 2×1, Number: 64	/	16×64
Convolution4 (C4)	Kernel Size: 3×1, Stride: 1×1, Number: 64	ReLU	16×64
Pooling4(P4)	Kernel Size: 2×1, Stride: 2×1, Number: 64	/	8×64
Convolution5 (C5)	Kernel Size: 3×1, Stride: 1×1, Number: 64	ReLU	6×64
Pooling5(P5)	Kernel Size: 2×1, Stride: 2×1, Number: 64	/	3×64
Fully-connected1/Flatten (F1)	/	/	192×1
Fully-connected2 (F2)	Weights: 192×100, Bias: 100×1	ReLU	100×1
Fully-connected3 (F3)	Weights: 100×4, Bias: 4×1	Softmax	4×1

By stacking convolutional layers and pooling layers in turn, we can extract high-level features, which need to be flattened into a 1-D vector, and then are fed into the fully-connected layer. For example, the output of the first fully-connected layer F_1 is obtained by flattening the feature $p_i^{P_5,D}$ of the pooling layer P_5 (as shown in Table 1) into a 1-D vector. The output of the fully-connected layer F_2 can be expressed as

$$x_i^{F_2,D} = \sigma_r(w^{F_2} x_i^{F_1,D} + b^{F_2}) \tag{20}$$

where $x_i^{F_1,D} = flatten(p_i^{P_5,D})$ is the flattened feature vector, w^{F_2} and b^{F_2} are the weights and biases of the fully-connected layer F_2.

The third fully-connected layer F_3 is employed for classification, which uses the softmax function to predict the labels of the source domain samples x_i^s and the target domain samples x_i^t. The output of the layer F_3 denotes the probability distribution of the labels for a sample. It can be expressed as follows.

$$\begin{aligned} d_i^{F_3,D} &= [P(y_i^D = 1 \left| x_i^{F_2,D} \right.) \cdots P(y_i^D = k \left| x_i^{F_2,D} \right.) \\ &\quad \cdots P(y_i^D = n \left| x_i^{F_2,D} \right.)]^{\mathrm{T}}, \\ P(y_i^D = k \left| x_i^{F_2,D} \right.) &= \frac{\exp((w_k^{F_3})^{\mathrm{T}} \cdot x_i^{F_2,D} + b^{F_3})}{\sum_{k=1}^{n} \exp((w_k^{F_3})^{\mathrm{T}} \cdot x_i^{F_2,D} + b^{F_3})} \end{aligned} \tag{21}$$

where $d_i^{F_3,D} = \left\{ d_i^{F_3,s}, d_i^{F_3,t} \right\}$ is the probability distribution of labels for the i-th couple of source-target samples. n represents the number of labels, $w_k^{F_3}$ and b^{F_3} are the weights and biases of the fully-connected layer F_3.

2) FULLY-CONNECTED LAYERS DOMAIN ADAPTATION

For a deep network, as the number of network layers deepens, the network becomes more and more dependent on specific tasks, while the shallow layers only learn the rough features. For different tasks, the shallow features are basically universal [28]. Inspired by this idea, we believe that it is more important to conduct the high-layer domain adaptation. Therefore, we perform domain adaptation on the three fully-connected layers. By using the MDDA method, the distribution discrepancy of the learned transferable features can be effectively reduced. Nevertheless, unlabeled samples in the target domain cannot be used to train the parameters of the fully-connected layer F_3. Therefore, it is necessary to introduce pseudo labels so as to solve this problem. The so-called pseudo label of the sample is to select the label with the maximum predicted probability as the approximate label. In the fully-connected layer F_3, we use the softmax function to predict the probability distribution of labels for the samples in the target domain. Combined with (21), the pseudo label can be obtained by the following equation.

$$\begin{aligned} \hat{y}_i^t &= [\hat{y}_k^t \cdots \hat{y}_k^t \cdots \hat{y}_k^t], \\ \hat{y}_k^t &= \begin{cases} 1, & \text{if } k = \arg\max_k d_i^{F_3,D} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{22}$$

where $\hat{y}_i^t$ is the pseudo label of the $i - th$ sample in the target domain.

3) TRAINING PROCEDURE

The proposed DTN with MDDA model is trained by jointly minimizing three types of losses: 1) the error between the predicted and true labels of the samples in the source domain, 2) the error between the predicted and pseudo labels of the samples in the target domain, 3) the fully-connected layers domain adaptation of the learned transferable features from cross-domain samples. In addition, we use mini-batch stochastic gradient descent (SGD) for network optimization. In other words, the MDDA is calculated between batches rather than the whole domains, which makes the practical calculation more easy and efficient. Therefore, the loss function including three regularization terms of the DTN with MDDA model is expressed as

$$L(\theta) = \min_{\theta} \frac{1}{m} \sum_{i=1}^{m} J(f(x_i^s), y_i^s) + \alpha \frac{1}{m} \sum_{i=1}^{m} J(f(x_i^t), \hat{y}_i^t) + \lambda D_{\mathcal{H}}(D_s, D_t) \tag{23}$$

where $J(\cdot, \cdot)$ is the cross-entropy loss function, $\theta = \{w, b\}$ is the parameter collection of the network, $f(\cdot)$ is the prediction function, $D_{\mathcal{H}}(D_s, D_t)$ represents the distribution discrepancy of the proposed MDDA, α and λ are the trade-off parameters, and m is the mini-batch number of the samples.

The gradient of the parameters can be calculated as:

$$\Delta_\theta = \frac{\partial J_s(\cdot, \cdot)}{\partial \theta} + \alpha \frac{\partial J_t(\cdot, \cdot)}{\partial \theta} + \lambda \frac{\partial D_{\mathcal{H}}(\cdot, \cdot)}{\partial \theta} \tag{24}$$

where Δ_θ is the mini-batch gradient operator to update the parameters of the network, $J_s(\cdot, \cdot)$ and $J_t(\cdot, \cdot)$ are the cross-entropy loss functions from source domain and target domain, respectively.

The flowchart of the training process for the DTN with MDDA model is presented in Fig. 4. In the domain partition stage, the vibration data sets collected from different machines are respectively considered as the source domain and target domain. For feature extraction, mini-batch samples from the source and target domains are fed into the DTN in order to obtain high-level transferable features. As for domain adaptation, the MMD of the learned transferable features is computed by (11), and then the pseudo labels for the target domain samples are predicted through (22). This process is the forward propagation (FP) of the network. We use (24) to calculate the minimum batch gradient so that the network parameters can be updated. This process is the backward propagation (BP) of the network. The proposed DTN with MDDA model is finally trained until the terminal conditions are satisfied. It should be noted that we propose to update μ and β after each epoch of iteration to avoid gradient explosion problems. In the fault diagnosis stage, the trained model is employed to classify samples from the target machine and output the diagnosis results.

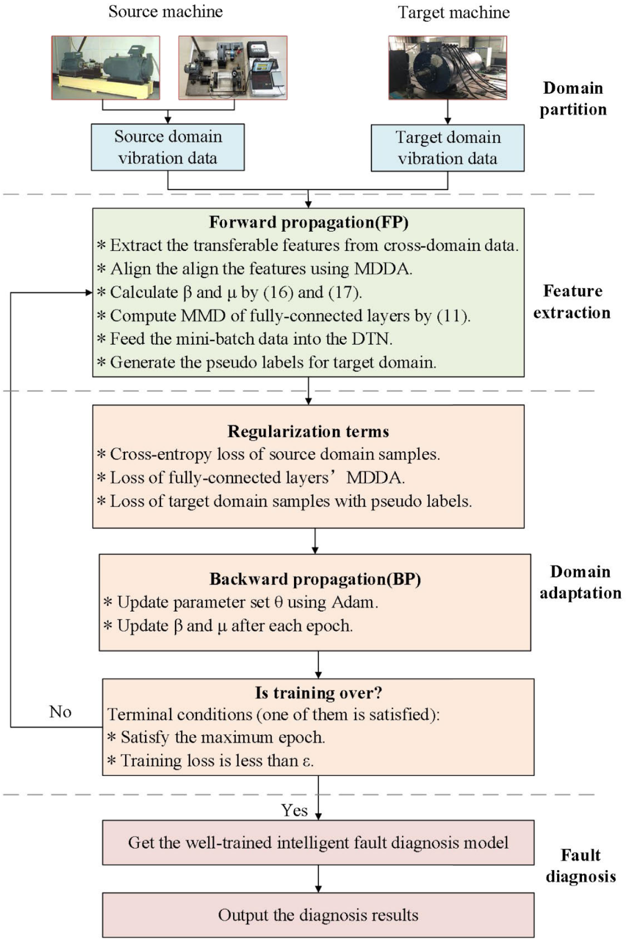

Source machine
Target machine
Domain partition
Source domain vibration data
Target domain vibration data
Forward propagation(FP)
* Extract the transferable features from cross-domain data.
* Align the align the features using MDDA.
* Calculate β and μ by (16) and (17).
* Compute MMD of fully-connected layers by (11).
* Feed the mini-batch data into the DTN.
* Generate the pseudo labels for target domain.
Feature extraction
Regularization terms
* Cross-entropy loss of source domain samples.
* Loss of fully-connected layers' MDDA.
* Loss of target domain samples with pseudo labels.
Backward propagation(BP)
* Update parameter set θ using Adam.
* Update β and μ after each epoch.
Domain adaptation
Is training over?
Terminal conditions (one of them is satisfied):
* Satisfy the maximum epoch.
* Training loss is less than ε.
No
Yes
Get the well-trained intelligent fault diagnosis model
Output the diagnosis results
Fault diagnosis

FIGURE 4. Flowchart of the training process for the proposed DTN with MDDA model.

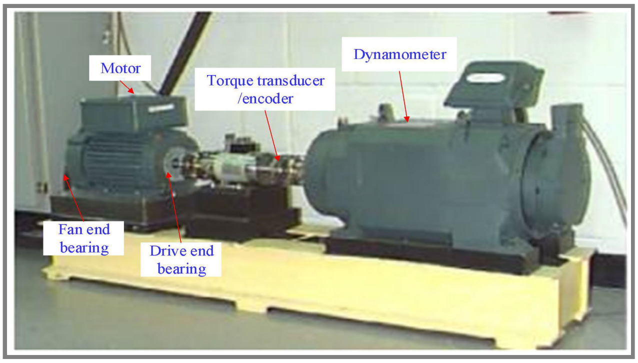


FIGURE 5. Motor driving mechanical system provided by CWRU.

IV. CASE STUDY

A. DATA DESCRIPTION

In this section, two laboratory bearing fault data sets and a wind turbine bearing fault data set are conducted to demonstrate the efficiency, superiority and practicability of the proposed DTN with MDDA model. The laboratory bearing data sets are labeled data, while the wind turbine bearing data set is unlabeled data. We try to identify the health states of wind turbine bearings by using the diagnosis knowledge from the laboratory bearings. The three data sets are described as below.

The first data set from the motor bearing are provided by Case Western Reserve University [29], as shown in Fig. 5. The vibration data from bearings (SKF6205) was collected using a accelerometer placed at the drive end of the motor, single point faults were introduced to the test bearings using electro-discharge machining, the health states of bearings includes normal (N), inner race fault (IF), ball fault (BF) and outer race fault (OF) (corresponding labels are 0∼3). In addition, it is worth noting that the fault diameter in each fault state was 0.014 inches, the sampling frequency was 12kHz. As shown in Table 2, the selected data set A was acquired

TABLE 2. Details of the data sets.

data sets	Bearing types	Fault categories	Labels	Number of samples	Operation conditions
A	SKF6205	N	0	4×100	0 HP (1797 rpm)
		IF	1		
		BF	2		
		OF	3		
B	SKF6205	N	0	4×100	3 HP (1730 rpm)
		IF	1		
		BF	2		
		OF	3		
C	N205	N	0	4×100	17.5 N(1500 rpm)
		IF	1		
		BF	2		
		OF	3		
D	NU1030	N	0	4×100	1590 rpm
		IF	1		
		BF	2		
		OF	3		

with load of 0HP (the motor speed was about 1797r/min), and the selected data set B with load of 3HP (the motor speed was about 1730 r/min). The data set A and B contain 400 samples respectively, and each sample has 1200 sampling points.

Another laboratory bearing data set was collected from a test rig, as shown in Fig. 6. The test rig consists of an AC drive motor, magnetic powder loader, test bearing, accelerometer, test system and other auxiliary parts, which can realize the vibration test of rolling bearings under different working conditions. The bearing type was N205, the driving speed was 1500rpm and the load was set to 17.5N. The vibration signal was collected by the accelerometer, the sampling frequency was set to 12.8kHz, and man-made electrical cutting damages was conducted in different positions of the testing bearing, i.e., N, IF, BF, and OF, as shown in Table 2. There are 400 samples in the data set C, and each sample has 1200 sampling points.

The other data set was collected from a real-world wind turbine bearing, which was installed on the test bench, as shown in Fig. 7. In the experiment, the rotation speed of the generator was 1590rpm. According to the experimental standard of wind turbine [30], the fault signal of the testing bearing was collected by a accelerometer, the sampling frequency was 12.8kHz, as shown in Table 2. There are four fault types and 400 samples in the data set D, and each sample has 1200 sampling points.

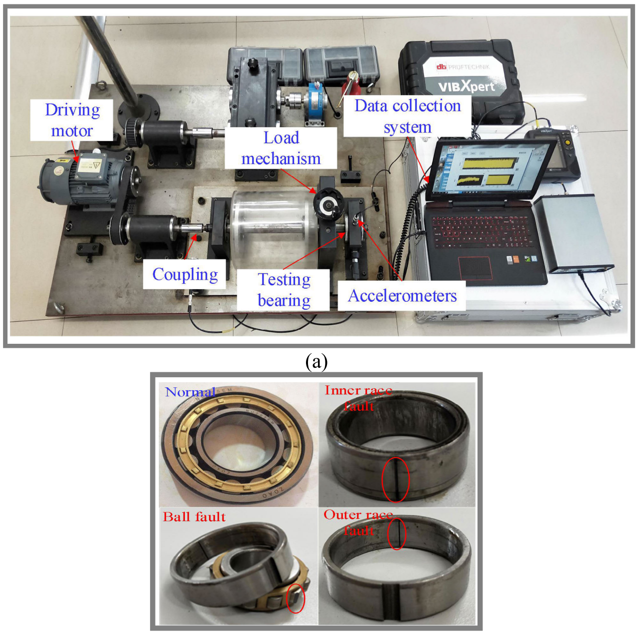
According to Table 2, we obtain three transfer learning tasks, which are A → D, B → D and C → D. The data sets A, B, and C are regarded as the source domain, which has labeled samples, while the data set D is viewed as the target domain, which requires us to label the samples, and the goal of the tasks is to make the classification results close to the true value as much as possible.

B. CASE 1: TRANSFER TASKS FROM A → D AND B → D

1) TRANSFER RESULTS OF THE PROPOSED METHOD

In the proposed method, μ and β are two important weighting factors, and their different values will directly affect the transfer learning results of the model. Therefore, it is necessary to analyze the parameter selection. In order to verify our estimation, we record the performance of MDDA by searching different values of μ and β, that is, better values of μ and β contribute better transfer performance. In the presented DTN, the classification accuracy depends on the learned features in the fully-connected layers F_3, which is the highest-level features before classification. Thus, the learned transferable features in the layer F_3 is utilized to analyze transfer results of DTN after selecting different parameters. And then we run MDDA by searching μ and $\beta \in \{0, 0.1, \cdots, 0.9, 1.0\}$. As shown in Fig. 8, we draw the results of MDDA under different values of μ and β.

It can be seen from Fig. 8(a) and Fig. 8(b), on the one hand, the classification accuracy obviously varies with



(b)

FIGURE 6. The laboratory bearing test rig: (a) Schematic diagram of the test rig; (b) The damaged positions.

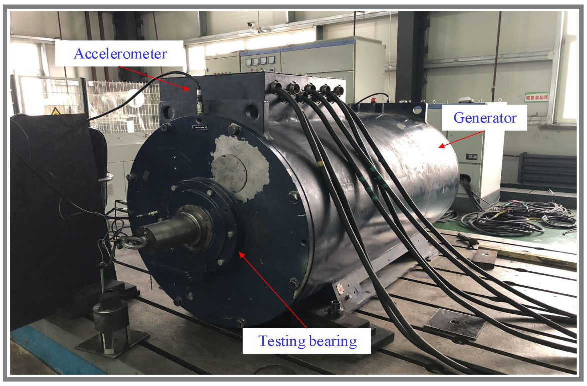


FIGURE 7. The wind turbine bearing test bench in real-world.

different choices of μ and β. This indicates that it is necessary to consider both the different effects between marginal and conditional distributions, and the effects of different kernel functions in transfer learning. On the other hand, we can also observe that the optimal values of μ and β vary with different transfer tasks. Thus, it is necessary to dynamically adjust the two parameter values according to different tasks. Moreover, for a given task, the optimal values of μ and β may not be unique, that is, the classification results for different μ and β may be the same.

Since the optimal μ and β are not unique, we can not directly compare the optimal values with the estimated values. Instead, we compare the performances (classification accuracy) achieved by the optimal values and the estimated values, as shown in Table 3.

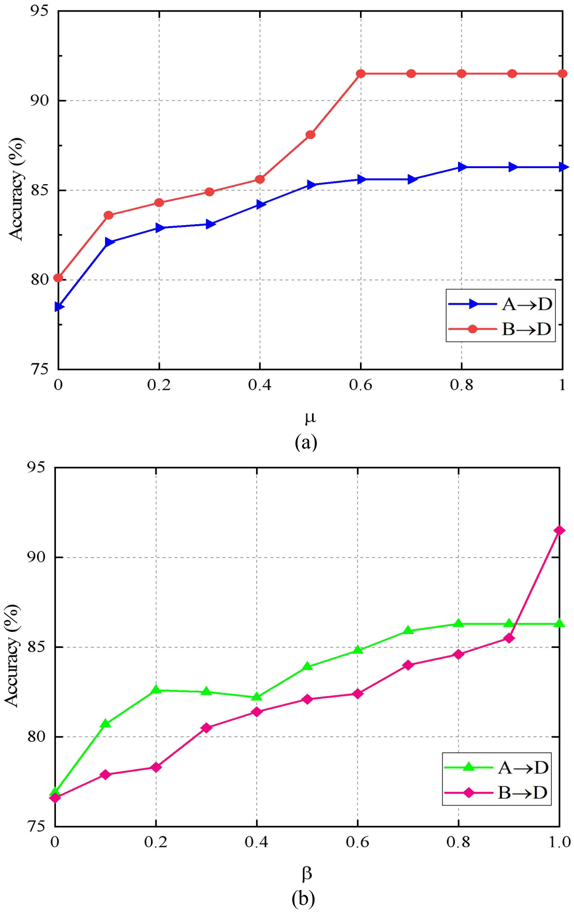


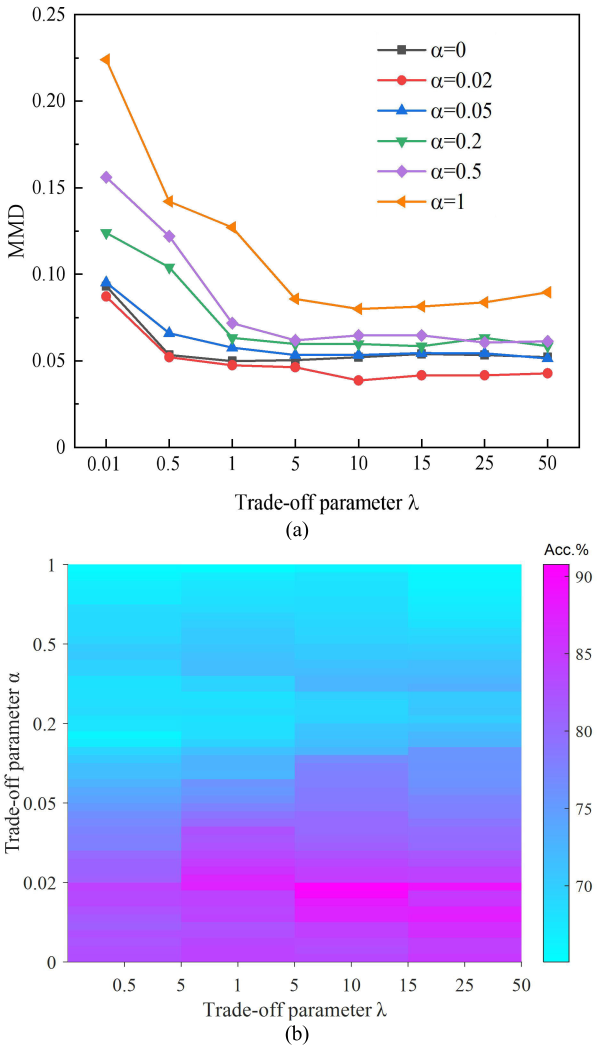
FIGURE 8. Performance of the two tasks when searching the optimal values of μ and β: (a) Transfer results of different μ; (b) Transfer results of different β.

TABLE 3. Comparison of Model Parameters on Different Transfer Task.

Tasks	A→D	B→D
μ_{opt}	85.62%	86.69%
$\hat{\mu}$	85.34%	86.81%
Performance variation	-0.28%	+0.12%
β_{opt}	87.42%	89.53%
$\hat{\beta}$	87.41%	89.94%
Performance variation	-0.01%	+0.41%

We list the classification results corresponding to the estimated value and the true value. These results clearly show that the classification accuracy of our quantitative evaluation of the adaptive factor μ and the balance factor β is extremely close to the results from grid search. In practical applications, the two factors need to be recalculated after each iteration, which means our estimation is more effective.

It is worth noting that α and λ are also two important trade-off parameters that affect classification accuracy. Take the task B $\rightarrow$ D as an example, the parameter α is searched from $\{0, 0.02, 0.05, 0.2, 0.5, 1\}$, and the parameter λ is chosen from {0.01, 0.5, 1, 5, 10, 15, 25, 50}. Each experiment is performed 10 times and the average value is calculated, as shown in Fig. 9.



(a)

(b)

FIGURE 9. Transfer learning results with different trade-off parameters for the task B → D: (a) MMD of the learned transferable features, (b) Classification accuracy on the data set D.

It can be seen from Fig. 9(a), the MMD of the learned transferable features is significantly reduced when the parameters α and λ are set as 0.02 and 10, respectively. After training the DTN model with different trade-off parameters, the classification accuracy on the data set D is shown in Fig. 9(b). When the parameter α is smaller than 0.05 and the parameter λ exceeds 5, the classification accuracy ranges from 80% to 90%. Specifically, the classification accuracy reaches its maximum value when α and λ the classification accuracy reaches the maximum value when α and λ are respectively set as 0.02 and 10. It demonstrates that the classification accuracy changes with the MMD of the transferable features. The smaller the MMD value, the higher the accuracy.

2) COMPARISONS WITH OTHER METHODS

We compare the transfer results and transfer performances of the presented method with those state-of-the-art machine learning methods including CNN, transfer component analysis (TCA), joint distribution adaptation (JDA) [31], geodesic flow kernel (GFK) [32], deep domain confusion (DDC), deep adaptation network (DAN) and DTN with DDA. For comparison, the structural parameters of CNN are the same as DTN. TCA is a commonly used transfer learning method, which introduces the idea of principal component analysis into transfer learning and only adapts the marginal distribution without considering the conditional distribution. JDA considers jointly adapting the marginal distribution and conditional distribution, and assumes that both distributions are equally important. GFK is a manifold feature extraction method that replaces Euclidean distance with geodesic distance. DDC aims at the AlexNet network, adding an adaptation layer before the classification to reduce the MMD distance between source and target domains. DAN is the improved version of DDC, in which multi-layer adaptation is performed in the fully connected layers, and multi-kernel MMD is used to replace single-kernel MMD. The inputs of TCA and JDA are frequency spectrum data, while the inputs of other methods are raw vibration data. Note that the optimal parameters are selected for each method in the experiments (more details shown in the Appendix). Moreover, ten trials are conducted, and the mean of each method is listed in Table 4.

The average classification accuracy of the proposed DTN with MDDA method is 88.9%, which is the highest one among the eight methods. Due to the absence of domain adaptation, the distribution discrepancy of the source and target domains is large, the average accuracy of CNN only reaches 66.4%, which is smaller than the accuracy achieved by the proposed model. TCA, JDA, and GFK are shallow transfer learning methods that can only extract lower-level features, and conduct the unsupervised domain adaptation, so their average accuracies are poorer than our model. This indicates that they are not suitable for dealing with the cross-machine tasks subject to serious distribution discrepancy. DDC and DAN are deep transfer learning methods, so their average accuracies are better than the previous methods. However, they only reduce the distribution discrepancy by minimizing the average distance of the transferable features, they reach lower accuracies than the proposed method. DTN with DDA method evaluates the relative importance of the marginal distribution and the conditional distribution, but ignores the phenomenon that different kernel functions have different geometric metrics for features. For cross-machine transfer learning tasks, the mapping effect of a single kernel function is not ideal. Although the average accuracy of DTN with DDA reaches 80.9%, it is still lower than our method. In addition, transfer ratio (TR) [33] is introduced to compare the transfer performance of the presented method with that of other methods. It is defined

TABLE 4. Classification accuracy (%) and transfer performance of different methods for different transfer task.

Methods	Input	A→D	B→D	Average	Transfer ratio
CNN	Raw vibration data	66.9	65.8	66.4	0.53
TCA	Frequency spectrum data	64.1	62.7	63.4	0.51
JDA	Frequency spectrum data	68.6	69.5	69.1	0.62
GFK	Raw vibration data	65.6	66.3	65.9	0.61
DDC	Raw vibration data	75.6	76.5	76.1	0.76
DAN	Raw vibration data	78.6	80.5	79.6	0.78
DTN.w.DDA	Raw vibration data	79.7	82.0	80.9	0.81
DTN.w.MDDA （ours）	Raw vibration data	86.3	91.5	88.9	0.88

as follows:

$$TR = \frac{1}{p}\sum_{i=1}^{p}[1-err(S_i,T_i)]/[1-err(T_i,T_i)] \qquad (25)$$

where $err(S_i, T_i)$ denotes the transfer error between the source and the target domains, $err(T_i, T_i)$ represents the testing error in the target domain, p is the number of transfer learning tasks.

Transfer ratio is used to comprehensively evaluate the transfer performance of a method for different transfer learning tasks. The higher the transfer ratio, the better the transfer performance that a method obtains. As shown in Table 4, the transfer ratio of the proposed DTN with MDDA method is 0.88, which is the highest one among the eight methods. It shows that our method has superior transfer performance.

In order to give a clear and intuitive understanding of the transfer learning process, the t-distributed stochastic neighbor embedding (t-SNE) algorithm [34] is used for network visualization. This algorithm can reduce the dimension of the transferable features so that the distribution of the features can be illustrated in the form of 2-D image. Taking the B → D transfer learning task as an example, for comparison, the transferable features learned by our method and other methods via t-SNE are respectively shown in Fig. 10(a)∼(h).

From Fig. 10(a), we observe that the transferable features learned by CNN exist serious distribution divergence, which makes the features of the target domain get poor clustering results and small inter-class distances. Therefore, when the model is only trained by the source domain samples, CNN cannot effectively classify unlabeled samples in the target domain. From Fig. 10(b), (c) and (d), the distribution of transferable features is not effectively aligned using TCA, JDA and GFK. Moreover, the transferable features of the samples under the normal state and outer race fault, and the samples under inner race fault and ball fault are not clearly separated, which shows that the distribution discrepancy is still serious and the average diagnosis results of data set D are listed in Table 5.

TABLE 5. Classification results and transfer performance for the transfer learning task C → D.

Methods	Input	Accuracy(%)	Transfer ratio
CNN	Raw vibration data	48.5	0.48
TCA	Frequency spectrum data	41.7	0.42
JDA	Frequency spectrum data	44.0	0.45
GFK	Raw vibration data	42.3	0.43
DDC	Raw vibration data	58.4	0.59
DAN	Raw vibration data	63.1	0.64
DTN.w.DDA	Raw vibration data	75.9	0.76
DTN.w.MDDA （ours）	Raw vibration data	81.8	0.82

For Fig. 10(e) and (f), DDC and DAN minimize the average distance of the extracted transferable features between the source and target domains before classification, which makes the cross-domain distribution discrepancy significantly reduced by supervised learning. Thus, DDC and DAN obtain higher diagnosis accuracy compared with CNN, TCA, JDA and GFK. For Fig. 10(g), DDA can dynamically align marginal distribution and conditional distributions of the learned features according to the specific situation, which enables the cross-domain distribution to be effectively adapted, thereby greatly reducing the distribution divergence. Fig. 10(h) shows that the proposed DTN with MDDA method not only can extract higher-level transferable features, but also dynamically adjust the feature

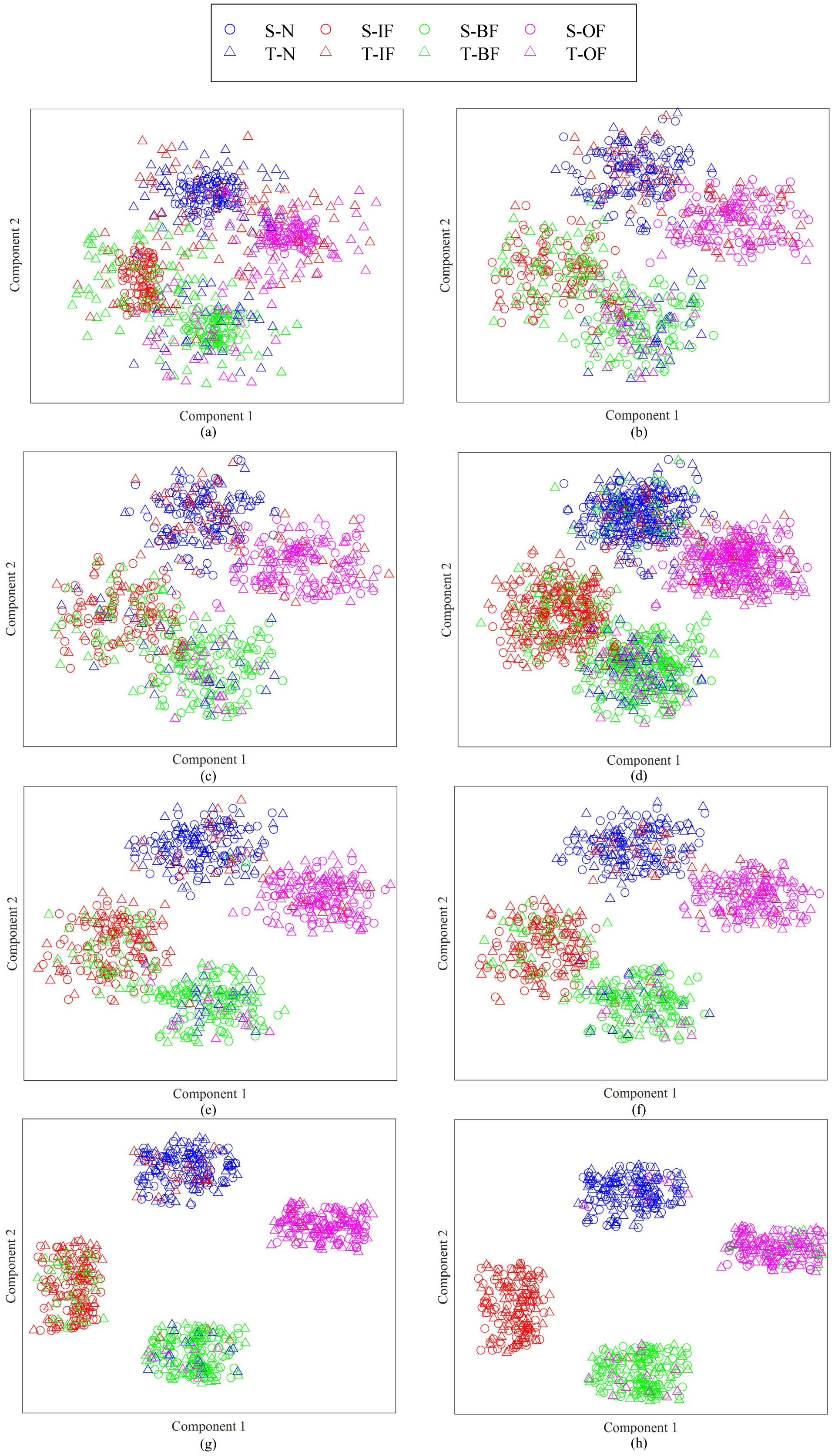


FIGURE 10. The visualization of the learned features for the transfer learning task B → D: (a) CNN, (b) TCA, (c) JDA, (d) GFK, (e) DDC, (f) DAN (g) DTN.w.DDA, and (h) DTN.w.MDDA.

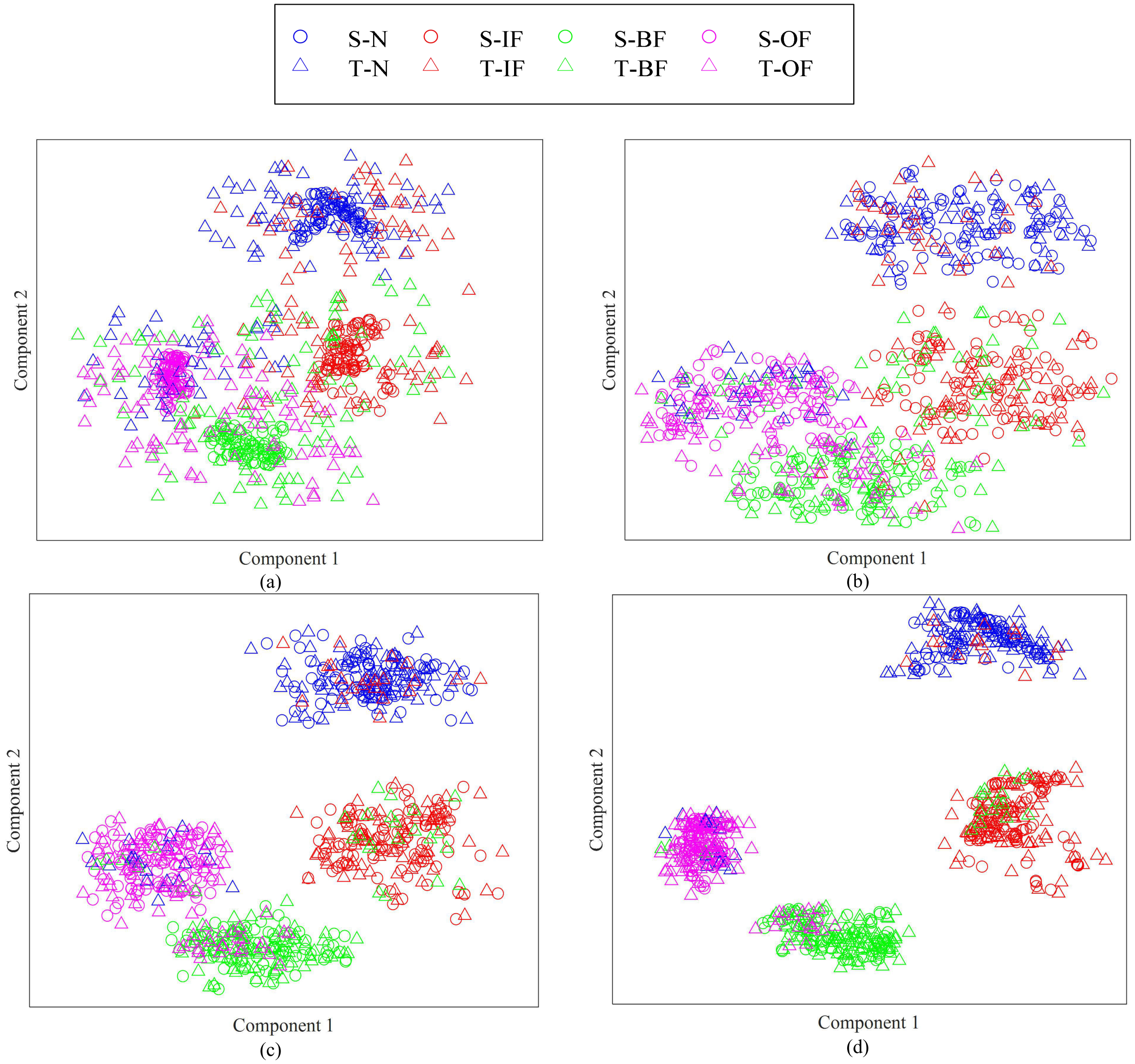


FIGURE 11. The visualization of the learned features for the transfer learning task C → D: (a) CNN, (b) DAN, (c) DTN.w.DDA, (d) DTN.w.MDDA.

representation ability of different kernel functions and distribution divergence between two domains simultaneously. Therefore, the goal of maximize the between-class distance and minimize the within-class distance is achieved, so that the target domain samples are correctly classified. The visualization of these results gives a good explanation why the proposed method shows higher classification accuracy than other methods.

C. CASE 2: TRANSFER TASK FROM C → D

The effectiveness of the presented method can also be validated by another case. In the case, the diagnosis knowledge of data set C is utilized to identify the health states of the wind turbine bearings, i.e., the transfer learning task C → D. Follow the previous approach, 10 trails are performed, According to the results in Table 5, the proposed DTN with MDDA model obtains the average accuracy of 81.8%. The transfer performance of the model for the task C → D is also measured by the transfer rate, and its value is 0.82. To compare the transfer results and transfer performance of the proposed method with other seven methods adopted in Case 1, for each method, the experiments are conducted under the optimal parameter selection. From the results listed in Table 5, for the transfer learning task C → D, the proposed method still has the highest diagnosis accuracy and transfer rate among the eight methods, which shows that the DTN with MDDA method has superior performance compared with other methods.

The learned transferable features using CNN, DAN, DTN with DDA, DTN with MDDA are illustrated in Fig. 11 via

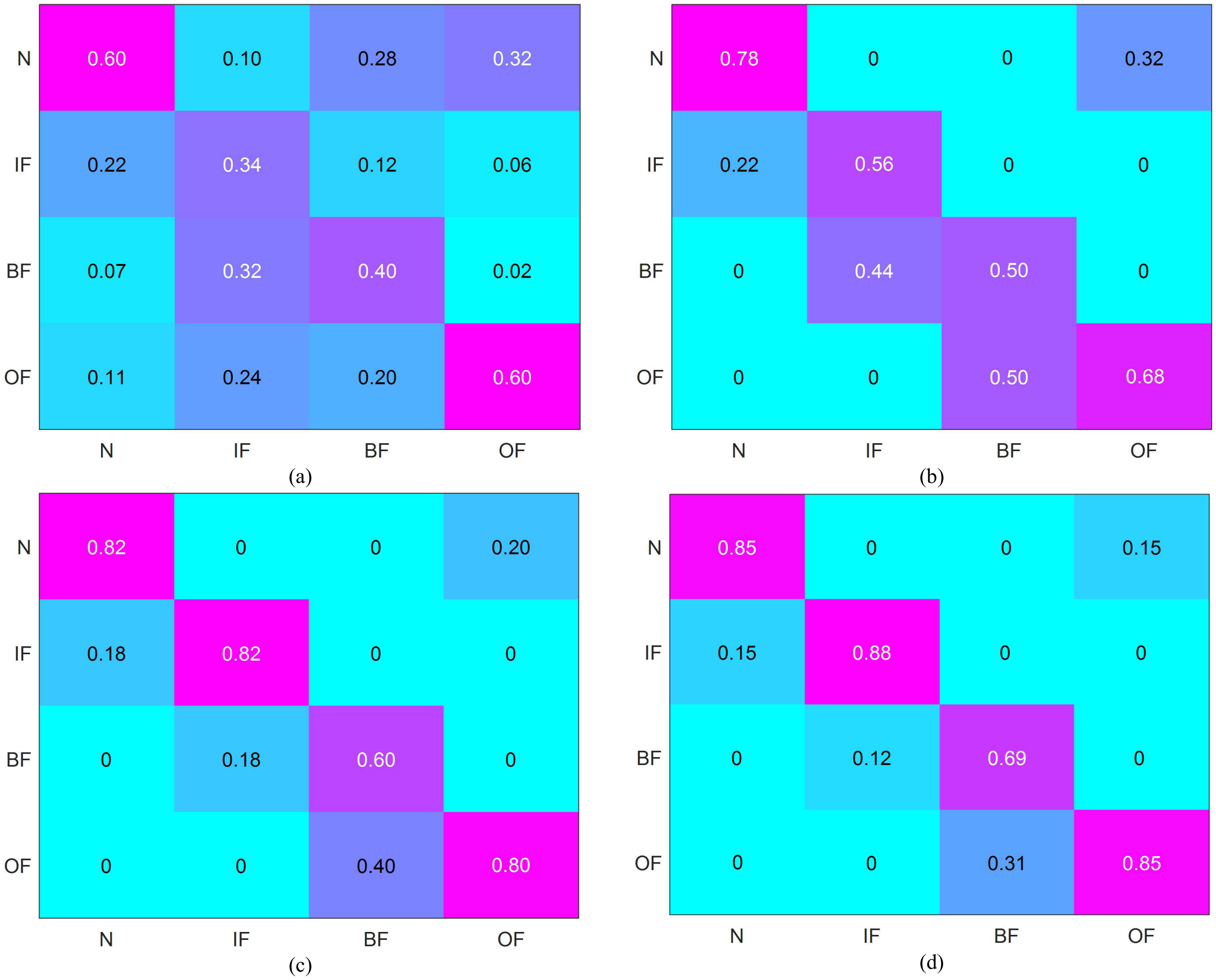


FIGURE 12. The confusion matrix for the transfer learning task C → D: (a) CNN; (b) DAN; (c) DTN.w.DDA; (d) DTN.w.MDDA.

t-SNE algorithm. According to the visualization results, we also discussed the confusion matrix for classification results of the data set D under the four methods, as shown in Fig. 12.

From Fig. 11(a), the transferable features learned by the CNN exist serious distribution divergence. Consequently, when the diagnosis knowledge of data set C is transferred to data set D, the classification accuracy of CNN for data set D only reaches 48.5%, as shown in Fig. 12. (a). As for DAN, due to the multi-layer domain adaptation before classification, as a result, the cross-domain distribution discrepancy is reduced to some extend by minimizing MMD of the learned high-level transferable features, as shown in Fig. 11(b). Additionally, the classification result of DAN is higher than that of CNN for data set D, as shown in Fig. 12(b). For DTN with DDA, the relative importance of the marginal and conditional distributions is evaluated, the distribution discrepancy of cross-machine samples is further reduced, as shown in Fig. 11(c). Thus, the classification result of DTN with DDA is also improved for data set D, as shown in Fig. 12(c). From Fig. 11(d), the proposed DTN with MDDA method not only dynamically adapts the distribution discrepancy, but also obtains the suitable feature mapping kernel function, and more comprehensively demonstrates the information of different transferable features. Therefore, the proposed method still has higher classification accuracy and better transfer performance than other methods. Furthermore, although the transferability of different sub-category is different, the classification result of each sub-category is able to be corrected by our model, as shown in Fig. 12(d).

D. DISCUSSION UNDER NOISE ENVIRONMENT

In real industries, environmental noise is inevitable, and these noises may greatly affect the results of transfer learning fault diagnosis. In this section, the transfer performance of the proposed method is estimated when the target machine is under additional environmental noise. Gaussian white noise is added to the original vibration data. The signal-to-noise (SNR) of the noise data is defined as follows:

$$SNR(dB) = 10log_{10}(P_{signal}/P_{noise}) \tag{26}$$

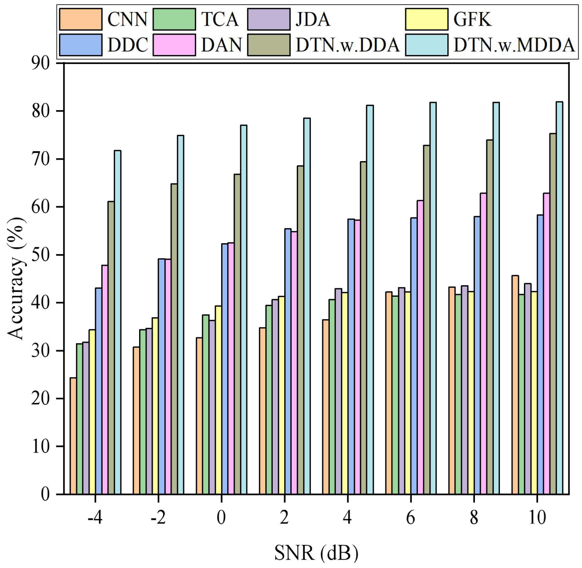


FIGURE 13. Diagnosis results with different levels of additional noises for the transfer learning task C → D.

where P_{signal} and P_{noise} represent the powers of the original vibration signal and the additional Gaussian white noise respectively. Next, select the SNR of $-4\sim 10$dB for evaluation, and then test the model with noisy data.

The classification accuracy of the transfer learning task C → D after adding white noise is demonstrated in Fig. 13. It can be observed that the classification accuracy of each method declines to some extent, which indicates that the performance of cross-machine fault diagnosis is obviously affected by environmental noise. However, comparing the eight methods, the classification accuracy of our model is significantly higher than the other methods (DTN with DDA reaches 60∼75%, DTN with MDDA reaches 70∼82%). This is because the proposed network structure has the wide first-layer convolution kernel and the deep small convolutional layers, which can more effectively suppress high-frequency noise. Thus, the proposed model can remain robust within a certain range even working under noisy signals. It can be seen from Fig. 13, the classification accuracy and robustness of the DTN with MDDA model are superior compared with other methods. The results shows that our model performs well under noisy environment conditions without any denoising pre-processing.

V. ANALYSIS AND DISCUSSTION

(1) In the training process of the DTN with MDDA model, the available data collected from the wind turbine bearing is unlabeled. Such scenario tallies with the actual situation, in which insufficient labeled data or even no labeled data used to adequately train an intelligent diagnosis model. The DTN with MDDA model can extract transferable features to reduce the distribution discrepancy between data sets from different machines.

The similarity of feature distribution greatly affects the accuracy of cross-machine fault recognition. Therefore, the proposed model is a promising method to complete cross-machine fault diagnosis. When there is no label available in the real world, according to the results shown in Table 4 and Table 5, the proposed method can obtain higher diagnosis accuracy and better transfer performance than other commonly used methods. In fact, the fault diagnosis of bearings is just a case study, and the presented method can also be employed for fault diagnosis of other machine components, such as gearboxes, motors and ball screws.

(2) It is noted that, as shown in Fig. 10 and 11, although the CNN model performs well in traditional fault diagnosis, it has lower accuracy in transfer learning. The reason for this phenomenon is that there are large domain shift in the source and target domains.

Therefore, domain adaptation must be carried out in transfer learning. In addition, shallow transfer learning methods, such as TCA, JDA, and GFK, are easier to adjust hyperparameters than deep transfer learning methods. However, due to in absent of supervised learning, the diagnosis accuracy of shallow transfer learning methods is not as good as deep transfer learning methods. Moreover, the traditional deep transfer learning methods such as DDC and DAN only use the loss function to minimize the distribution distance, but do not consider the relative importance of the marginal and conditional distributions in different data sets, which leads to the unsatisfactory classification accuracy obtained for cross-machine diagnosis. The DDA method dynamically evaluates the relative importance of the two distributions in different data sets. However, for different transferable features, DDA method does not consider the mapping capabilities of different kernel functions. The above analysis shows that the proposed MDDA method is an domain adaptation method with relatively strong comprehensive ability. When we embed MDDA into a deep learning model, the novel framework can effectively solve the problems of low accuracy and poor transfer performance for cross-machine fault diagnosis.

(3) The proposed DTN model has a special architecture with wide first-layer convolution kernel and several deep small convolutional layers, so that the learning ability of the model is very excellent even under noisy environment conditions, as shown in Fig. 13. This indicates that the proposed model is more suitable for real data under different background noises than the conventional CNN model. The model is worth popularizing in real industrial application.

VI. CONCLUSION

In this paper, a new intelligent fault diagnosis method based on transfer learning named DTN with MDDA is presented. The potential relationship between different but related mechanical components is mined, and cross-domain transferable diagnosis knowledge is developed. In the proposed method, a deep transfer network (DTN) is used to simultaneously extract transferable features from the source and target

TABLE 6. Detailed parameter settings for comparison experiments in Table 4.

Methods	Parameter settings
CNN	There are 5 convolution layers, 5 pooling layers, 3 fully-connected layers. Each of them with node numbers of 128, 64, 64, 32, 32, 16, 16, 8, 6, 3, 192, 100 and 4 from the input end to the output end, respectively. Softmax is used as the classifier.
TCA	Gaussian kernel is implemented, the optimal reduction dimension is searched from {2, 4, 8, 16, 32, 64, 128}. KNN is used as the classifier, the nearest neighbor is set as 1;
JDA	The parameters are the same as TCA.
GFK	The optimal dimension of geodesic flow kernel is selected by subspace disagreement measure (SDM). KNN is used as the classifier, the nearest neighbor is set as 1;
DDC	There are 8 layers AlexNet network structure, including 5 convolution layers, 3 fully-connected layers. Each of them with node numbers of 128, 64, 32, 16, 6, 192, 100 and 4 from the input end to the output end, respectively.The MMD distance is added to the 7-th layer (feature layer, the upper layer of softmax) to reduce the discrepancy between source and target domain.
DAN	The network structure is the same as DDC.The last two layers are attached with MK-MMD losses.
DTN.w.DDA	The network structure of DTN is shown in Table 1, DDA method is used as the fully-connected layers domain adaptation.
DTN.w.MDDA (ours)	The network structure of DTN is shown in Table 1, MDDA method is used as the fully-connected layers domain adaptation.

domains. Additionally, the multi-kernel dynamic distribution adaptation (MDDA) method constructs a weighted mixed kernel function, which combines the advantages of different kernel functions to map the transferable features to a unified feature space. It also can dynamically evaluate the relative importance of marginal distribution and conditional distribution, which effectively reduces the distribution discrepancy of the source and target domains. The proposed method improves the transfer ability of diagnosis knowledge between different machines. Three transfer learning tasks are used to verify the proposed method. The results show that when unlabeled data in the target domain is acquired, the proposed DTN with MDDA method can identify the health states of real wind turbine bearings and maintain good robustness under noisy data.

In addition, compared with other state-of-the-art methods, the proposed method achieves higher classification accuracy and better transfer performance, and has a promising industrial application prospect in cross-machine fault diagnosis.

In the future, we plan to expand the DTN with MDDA method to more realistic scenarios, such as online transfer learning, and apply it to more complex fault diagnosis, such as cross-component fault diagnosis from bearings to gearboxes.

APPENDIX

Adam optimizer is used to train all the deep models (CNN, DDC, DAN, DTN), batch size is 64, learning rate is 0.0001, the number of iterations is 20000. The parameter settings can be shown in Table 6.

REFERENCES

[1] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to RUL prediction," *Mech. Syst. Signal Process.*, vol. 104, pp. 799–834, May 2018, doi: 10.1016/j.ymssp.2017.11.016.

[2] X. Li, X.-D. Jia, W. Zhang, H. Ma, Z. Luo, and X. Li, "Intelligent cross-machine fault diagnosis approach with deep auto-encoder and domain adaptation," *Neurocomputing*, vol. 383, pp. 235–247, Mar. 2020, doi: 10.1016/j.neucom.2019.12.033.

[3] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, "Deep model based domain adaptation for fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2296–2305, Mar. 2017, doi: 10.1109/TIE.2016.2627020.

[4] Z. Luo, J. Wang, R. Tang, and D. Wang, "Research on vibration performance of the nonlinear combined support-flexible rotor system," *Nonlinear Dyn.*, vol. 98, no. 1, pp. 113–128, Oct. 2019, doi: 10.1007/s11071-019-05176-2.

[5] C. Paroissin, "Inference for the Wiener process with random initiation time," *IEEE Trans. Rel.*, vol. 65, no. 1, pp. 147–157, Mar. 2016, doi: 10.1109/TR.2015.2456056.

[6] W. Peng, Z.-S. Ye, and N. Chen, "Joint online RUL prediction for multivariate deteriorating systems," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 2870–2878, May 2019, doi: 10.1109/TII.2018.2869429.

[7] J. Li, X. Li, D. He, and Y. Qu, "A domain adaptation model for early gear pitting fault diagnosis based on deep transfer learning network," *Proc. Inst. Mech. Eng., O, J. Risk Rel.*, vol. 234, no. 1, pp. 168–182, Feb. 2020, doi: 10.1177/1748006X19867776.

[8] J. Zhu, N. Chen, and C. Shen, "A new deep transfer learning method for bearing fault diagnosis under different working conditions," *IEEE Sensors J.*, vol. 20, no. 15, pp. 8394–8402, Aug. 2020, doi: 10.1109/JSEN.2019.2936932.

[9] W. Qian, S. Li, P. Yi, and K. Zhang, "A novel transfer learning method for robust fault diagnosis of rotating machines under variable working conditions," *Measurement*, vol. 138, pp. 514–525, May 2019, doi: 10.1016/j.measurement.2019.02.073.

[10] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors*, vol. 17, no. 2, p. 425, Feb. 2017, doi: 10.3390/s17020425.

[11] B. Yang, Y. Lei, F. Jia, and S. Xing, "An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings," *Mech. Syst. Signal Process.*, vol. 122, pp. 692–706, May 2019, doi: 10.1016/j.ymssp.2018.12.051.

[12] X. Wang, C. Shen, M. Xia, D. Wang, J. Zhu, and Z. Zhu, "Multi-scale deep intra-class transfer learning for bearing fault diagnosis," *Rel. Eng. Syst. Saf.*, vol. 202, Oct. 2020, Art. no. 107050, doi: 10.1016/j.ress.2020.107050.

[13] J. Chatterjee and N. Dethlefs, "Deep learning with knowledge transfer for explainable anomaly prediction in wind turbines," *Wind Energy*, vol. 23, no. 8, pp. 1693–1710, Aug. 2020, doi: 10.1002/we.2510.

[14] M. S. Krejca and C. Witt, "Lower bounds on the run time of the univariate marginal distribution algorithm on OneMax," *Theor. Comput. Sci.*, vol. 832, pp. 143–165, Sep. 2020, doi: 10.1016/j.tcs.2018.06.004.

[15] S. M. Papalexiou and D. Koutsoyiannis, "A global survey on the seasonal variation of the marginal distribution of daily precipitation," *Adv. Water Resour.*, vol. 94, pp. 131–145, Aug. 2016, doi: 10.1016/j.advwatres.2016.05.005.

[16] M. Belalia, T. Bouezmarni, and A. Leblanc, "Bernstein conditional density estimation with application to conditional distribution and regression functions," *J. Korean Stat. Soc.*, vol. 48, no. 3, pp. 356–383, Sep. 2019, doi: 10.1016/j.jkss.2019.05.005.

[17] Y. Yuan, Y. Li, Z. Zhu, R. Li, and X. Gu, "Adversarial joint domain adaptation of asymmetric feature mapping based on least squares distance," *Pattern Recognit. Lett.*, vol. 136, pp. 251–256, Aug. 2020, doi: 10.1016/j.patrec.2020.06.007.

[18] J. Zhang, J. Liu, B. Pan, and Z. Shi, ''Domain adaptation based on correlation subspace dynamic distribution alignment for remote sensing image scene classification,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 7920–7930, Nov. 2020, doi: 10.1109/TGRS.2020.2985072.

[19] J. Wang, Y. Chen, W. Feng, H. Yu, M. Huang, and Q. Yang, ''Transfer learning with dynamic distribution adaptation,'' *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 1, pp. 1–25, Feb. 2020, doi: 10.1145/3360309.

[20] J. Liu, M. Bai, N. Jiang, and D. Yu, ''Structural risk minimization of rough set-based classifier,'' *Soft Comput.*, vol. 24, no. 3, pp. 2049–2066, Feb. 2020, doi: 10.1007/s00500-019-04038-8.

[21] X. Jia, M. Zhao, Y. Di, Q. Yang, and J. Lee, ''Assessment of data suitability for machine prognosis using maximum mean discrepancy,'' *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5872–5881, Jul. 2018, doi: 10.1109/TIE.2017.2777383.

[22] N. Wang, X. Gao, L. Sun, and J. Li, ''Bayesian face sketch synthesis,'' *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1264–1274, Mar. 2017, doi: 10.1109/TIP.2017.2651375.

[23] T. Han, C. Liu, W. Yang, and D. Jiang, ''Deep transfer network with joint distribution adaptation: A new intelligent fault diagnosis framework for industry application,'' *ISA Trans.*, vol. 97, pp. 269–281, Feb. 2020, doi: 10.1016/j.isatra.2019.08.012.

[24] L. Yang and P. Zhong, ''Robust adaptation regularization based on within-class scatter for domain adaptation,'' *Neural Netw.*, vol. 124, pp. 60–74, Apr. 2020, doi: 10.1016/j.neunet.2020.01.009.

[25] N. Cao, Z. Jiang, J. Gao, and B. Cui, ''Bearing state recognition method based on transfer learning under different working conditions,'' *Sensors*, vol. 20, no. 1, p. 234, Dec. 2019, doi: 10.3390/s20010234.

[26] J. Zhang, Y. Sun, L. Guo, H. Gao, X. Hong, and H. Song, ''A new bearing fault diagnosis method based on modified convolutional neural networks,'' *Chin. J. Aeronaut.*, vol. 33, no. 2, pp. 439–447, Feb. 2020, doi: 10.1016/j.cja.2019.07.011.

[27] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, ''Real-time motor fault detection by 1-D convolutional neural networks,'' *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7067–7075, Nov. 2016, doi: 10.1109/TIE.2016.2582729.

[28] E. Othman, Y. Bazi, F. Melgani, H. Alhichri, N. Alajlan, and M. Zuair, ''Domain adaptation network for cross-scene classification,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4441–4456, Aug. 2017, doi: 10.1109/TGRS.2017.2692281.

[29] Y. Li, X. Wang, S. Si, and S. Huang, ''Entropy based fault classification using the case western reserve university data: A benchmark study,'' *IEEE Trans. Rel.*, vol. 69, no. 2, pp. 754–767, Jun. 2020, doi: 10.1109/TR.2019.2896240.

[30] J. Guo, J. Wu, S. Zhangi, J. Long, W. Chen, D. Cabrera, and C. Li, ''Generative transfer learning for intelligent fault diagnosis of the wind turbine gearbox,'' *Sensors*, vol. 20, no. 5, pp. 1361–1377, 2020, doi: 10.3390/s20051361.

[31] Z. Xu, H. Darong, G. Sun, and W. Yongchao, ''A fault diagnosis method based on improved adaptive filtering and joint distribution adaptation,'' *IEEE Access*, vol. 8, pp. 159683–159695, 2020, doi: 10.1109/ACCESS.2020.3020906.

[32] T. Liu, X. Zhang, and Y. Gu, ''Unsupervised cross-temporal classification of hyperspectral images with multiple geodesic flow kernel learning,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9688–9701, Dec. 2019, doi: 10.1109/TGRS.2019.2928562.

[33] F. Wang, Z. Wang, C. Li, Y. Xiao, S. Wu, and P. Pan, ''The rejuvenating effect in hot asphalt recycling by mortar transfer ratio and image analysis,'' *Materials*, vol. 10, no. 6, p. 574, May 2017, doi: 10.3390/ma10060574.

MINGZHU LV received the M.S. degree in control theory and control engineering from the Shenyang University of Technology, China, in 2006, where she is currently pursuing the Ph.D. degree with the School of Mechanical Engineering. She is currently an Associate Professor with the School of Automatic Control Engineering, Liaoning Equipment Manufacture College of Vocational Technology, China. Her current research interests include wind turbine fault diagnosis and remaining useful life prediction.

SHIXUN LIU received the M.S. degree in electric machines and electric apparatus from the Shenyang University of Technology, China, in 2006, where he is currently pursuing the Ph.D. degree with the School of Electrical Engineering. He is currently a Senior Engineer with the CQC (ShenYang) North Laboratory, China. His current research interests include mechanical and electrical products testing and new energy technology development.

XIAOMING SU received the Ph.D. degree from the School of Information Science and Engineering, Northeastern University, Shenyang, China. He is currently a Professor and a Ph.D. Supervisor with the School of Mechanical Engineering, Shenyang University of Technology, China. His current research interests include mathematics, automatic control, and industrial engineering. He is also the Vice Chairman of the Liaoning Provincial Mathematical Society.

CHANGZHENG CHEN received the Ph.D. degree from the School of Mechanical Engineering, Northeastern University, Shenyang, China. He is currently a Professor and a Ph.D. Supervisor with the School of Mechanical Engineering, Shenyang University of Technology, China. His current research interests include vibration, noise, and fault diagnosis. He is also the Executive Director of the Fault Diagnosis Professional Committee of the Chinese Vibration Engineering Society and the Liaoning Vibration Engineering Society, the Consultant of the Fault Diagnosis Center of the Chinese Mechanical Engineering Society, the Director of the Noise and Vibration Control Professional Committee, the Noise Control Expert of the Shenyang Environmental Protection Bureau, and the Shenyang Environmental Protection Industry Council Member.