


Deep Adversarial Subdomain Adaptation Network for Intelligent Fault Diagnosis

Yanxu Liu , Yu Wang , Senior Member, IEEE, Tommy W. S. Chow , Fellow, IEEE, and Baotong Li 

Abstract—Recently, domain adaptation has received extensive attention for solving intelligent fault diagnosis problems. It aims to reduce the distribution discrepancy between the source domain and target domain through learning domain-invariant features. However, most existing domain adaptation methods mainly focus on global domain adaptation and overlook subdomain adaptation, which results in the loss of fine-grained information and discriminative features. To address this problem, in this article, a deep adversarial subdomain adaptation network is proposed. This network aligns the relevant distributions of subdomains by minimizing the local maximum mean discrepancy loss of the same categories in the source domain and target domain. Under the constraints of global domain adaptation and subdomain adaptation, the distribution discrepancy is reduced from the domain and category levels. Four transfer tasks under different machine rotating speeds and six transfer tasks on different but related machines were used to evaluate the effectiveness of the proposed method. The results demonstrated the robustness and superiority of the proposed method over five other methods.

Index Terms—Adversarial domain adaptation, deep learning, intelligent diagnosis, subdomain adaptation.

I. INTRODUCTION

ROLLING bearings are a key but vulnerable component of rotating machines. The failure of rolling bearings could cause a significant or even catastrophic accident in many industrial systems. Hence, it is important to guarantee the high stability and safety of machines by conducting health monitoring and fault detection and diagnosis [1].

In the past several years, with the development of machine learning technology and industrial big data, many data-driven intelligent methods have been used to perform bearing fault

diagnosis. Compared with traditional diagnosis methods, intelligent methods can achieve end-to-end diagnosis with sufficient labeled data [2], [3]. Although intelligent diagnosis methods have achieved significant success, most of the methods that performed well are based on two major assumptions: 1) massive labeled data under normal and failure conditions and 2) training and test data are collected from the same distribution [4], [5]. However, these two assumptions do not always hold in practice. In most practical applications, it is difficult to collect massive failure data because machines are unlikely to be allowed to keep operating under failure or close-to-failure conditions [6]. Moreover, changes of operation conditions and working environments lead to the distribution discrepancy, which results in a reduction of generalization performance [7].

To solve the abovementioned issues, transfer learning was developed. The major merit of transfer learning is that it can rely on collecting labeled data with massive faulty information from a source domain, which can be different from the machine type under diagnostic investigation, to diagnose other machines or working conditions using unlabeled data (target domain) [8]. Specifically, the source domain consists of abundant labeled faulty data, which is available for supervised training, but the target domain is unlabeled. The main mechanism of domain adaptation is to transfer the knowledge learned from a labeled source domain to an unlabeled target domain. As a result, domain adaptation can reduce the effect of the distribution discrepancy through learning domain-invariant features, which are transferable between the source domain and target domain [9]. Currently, there are two popular approaches for domain adaptation: metric based and adversarial based [10]. Metric-based domain adaptation approaches, such as maximum mean discrepancy (MMD) and correlation alignment (CORAL), embed moment matching as a loss function into a convolutional neural network (CNN) or residual neural network [11]–[13]. Adversarial-based domain adaptation uses a domain discriminator to distinguish whether the extracted features come from the source domain or the target domain, and the feature extractor is trained to deceive the domain discriminator [10].

Most recently, domain adaptation has attracted a great amount of attention for enhancing the generalization capability of the fault diagnosis model. Wen *et al.* [14] designed a spare autoencoder as a feature extractor and minimized the MMD to reduce the distribution discrepancy across the two domains. Yang *et al.* [15] embedded multilayer MMD domain adaptation and pseudolabel learning into a feature-based transfer CNN to reduce the distribution discrepancy. Han *et al.* [16] proposed a deep adversarial CNN with an additional discriminative classifier together with an adversarial learning strategy to boost the generalization ability of the classification model across domains.

Manuscript received July 5, 2021; revised October 5, 2021, November 19, 2021, and December 24, 2021; accepted January 2, 2022. Date of publication January 11, 2022; date of current version June 13, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 51875437, Grant 62073272, and Grant 61633001. Paper no. TII-21-2802. (Corresponding authors: Yu Wang; Baotong Li.)

Yanxu Liu, Yu Wang, and Baotong Li are with the State Key Laboratory for Manufacturing and Systems Engineering, School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: yanxuli@stu.xjtu.edu.cn; ywang95@xjtu.edu.cn; baotong.me@mail.xjtu.edu.cn).

Tommy W. S. Chow is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong (e-mail: eetchow@cityu.edu.hk).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TII.2022.3141783>.

Digital Object Identifier 10.1109/TII.2022.3141783

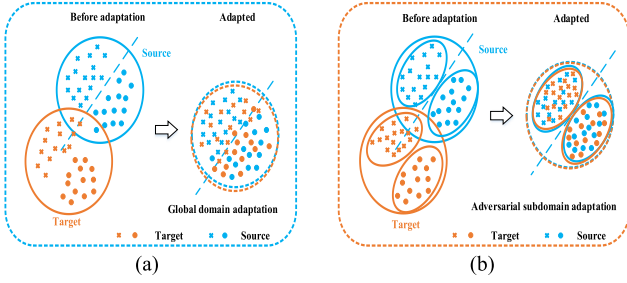


Fig. 1. (a) Global domain adaptation may lose fine-grained information. (b) Subdomain adaptation aligns the same category in both domains.

Guo *et al.* [17] proposed a 1-D deep convolutional transfer learning network (DCTLN) with condition recognition and domain adaptation to learn domain-invariant features between different machines. Wang *et al.* [18] used a supervised instance-based method to learn discriminative features to improve classification performance. They also introduced the Wasserstein distance as a domain discriminator to learn domain-invariant features.

Existing intelligent diagnosis methods based on domain adaptation mainly align the global distributions of the source and target domains. After global domain adaptation, the trained classifiers can theoretically classify both the source domain and target domain samples effectively [19], [20]. However, only aligning the global domain loses fine-grained information and mixes up discriminative structural features for each category. As shown in Fig. 1(a), the aforementioned problems lead to the misclassification of samples near the category decision boundary.

Recently, researchers have started to pay attention to subdomain adaptation, which can reduce the local domain shift [21], [22]. Specifically, the subdomain refers to a category in the source or target domain. It contains samples within the same faulty categories. Subdomain adaptation aligns the corresponding faulty categories in the two domains by learning domain-invariant and fault-discriminative features. In conclusion, global domain adaptation can align the global distributions at the domain level (big circles in Fig. 1), and subdomain adaptation can align the local distributions on category level (small circles in Fig. 1). After both global domain adaptation and subdomain adaptation, the learned domain-invariant features become interclass discriminative and intraclass compact, as shown in Fig. 1(b).

Motivated by the abovementioned idea, in this article, an intelligent cross-domain fault diagnosis method based on a deep adversarial subdomain adaptation network (DASAN) is proposed. The proposed DASAN model consists of three modules: feature extractor, domain discriminator, and label classifier. The feature extractor is designed to learn high-level feature representations of the raw input data. The domain discriminator aligns the global domain distributions by driving the feature extractor to generate cross-domain invariant features. The label classifier is trained to classify the source samples accurately and helps the feature extractor learn the discriminative features. Additionally, the local MMD (LMMD) is minimized to align the distribution of the corresponding subdomains for subdomain adaptation. Pseudolabel learning is also introduced to minimize the risk of

misclassifying the target domain samples as irrelevant subdomains. Finally, the proposed DASAN was evaluated on transfer tasks for different rotating speeds and different machines. A comprehensive comparison with several existing approaches demonstrates the effectiveness and superiority of DASAN. The main contributions of this article can be summarized as follows.

- 1) DASAN is proposed for intelligent fault diagnosis and provides a new perspective on solving domain adaptation problems. With subdomain adaptation, the learning mechanism and generalization performance can be improved significantly because of the inclusion of the fine-grained information from relevant categories.
- 2) A new distribution discrepancy measure (LMMD) is introduced to capture fine-grained information and preserve fault-discriminative structure of relevant subdomains. As a result, extracted invariant features can become interclass discriminative and intraclass compact.
- 3) A pseudolabel learning constraint is also introduced in DASAN to minimize the risk of misclassifying target samples as irrelevant samples.

The rest of this article is organized as follows. In Section II, the transfer learning problem is defined and the discrepancy metric of domain adaptation is introduced. In Section III, the details of the method, including the structure of DASAN, LMMD, and the training process, are provided. In Section IV, the transfer tasks are described. Finally, Section V concludes this article.

II. PRELIMINARIES

A. Problem Definition

In this article, we mainly focus on the intelligent fault diagnosis method based on unsupervised domain adaptation. We construct the source domain $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ with n_s labeled samples and target domain $D_t = \{(x_j^t)\}_{j=1}^{n_t}$ with n_t unlabeled samples. D_s and D_t follow the marginal probability distributions defined as p and q , respectively ($p \neq q$). We define the label space as $y = \{1, 2, 3, \dots, \mathbb{C}\}$, where \mathbb{C} is the type of health states (y_i^s is the corresponding label of x_i^s , $y_i^s = k (k \leq \mathbb{C})$ means that x_i^s belongs to the k th health state). In this article, we assume that the source and target domains share the same label space ($y^s = y^t$), which means that they have the same health states. We use the labeled source domain to train the classifier $f(\cdot)$ by minimizing the risk $R_s[f(\cdot); D_s \sim p] = \mathbb{E}_{(x_i^s, y_i^s)}[f(x_i^s) \neq y_i^s]$. In practice, the extracted features have an obvious discrepancy because the source domain and target domain have different distributions. Therefore, the trained source classifier cannot be applied on the target domain directly, which may result in many target samples being misclassified.

B. Discrepancy Metric

The discrepancy metric is a key component in domain adaptation. The MMD is the most frequently used discrepancy metric approach. The MMD and its extensions have been widely used to measure the distribution discrepancy between the source and target domain.

Given the fact that the source and target domains follow marginal probability distributions p and q , respectively, the

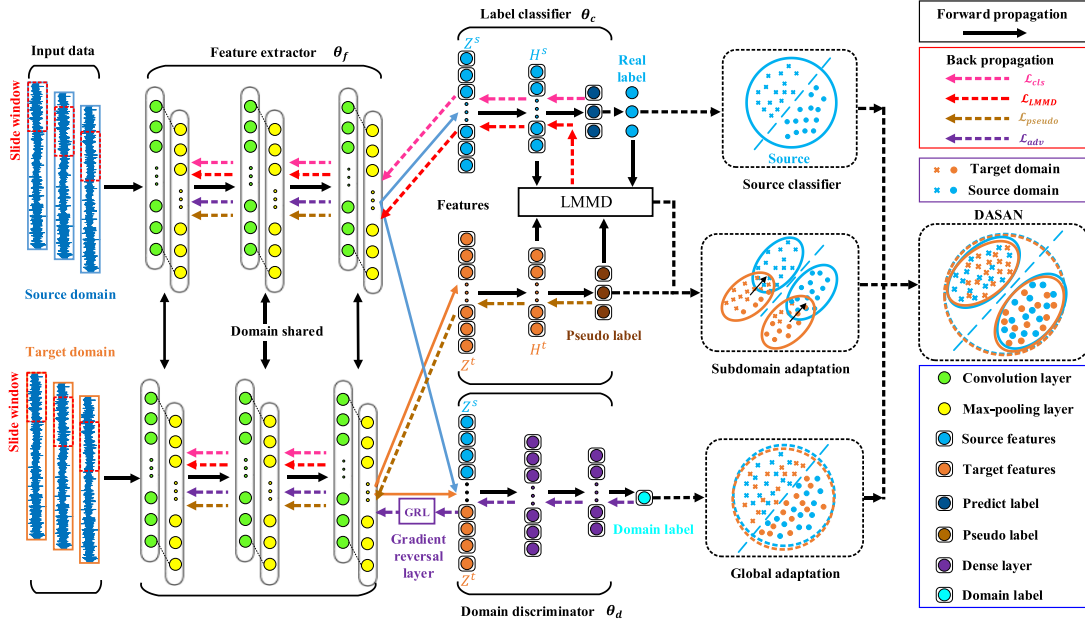


Fig. 2. Architecture of the proposed DASAN. Feature extractor with parameters θ_f , label classifier with parameters θ_c , and domain discriminator with parameters θ_d .

MMD measures the global distribution discrepancy between the mean embedding of the source and target domains in the reproducing kernel Hilbert space (RKHS). The RKHS is endowed with a characteristic kernel k that is defined as \mathcal{H} . Formally, the MMD is defined as follows [23]

$$\mathcal{L}_{\text{MMD}}(p, q) \triangleq \|E_p[\phi(x^s)] - E_q[\phi(x^t)]\|_{\mathcal{H}}^2 \quad (1)$$

where $\phi(\cdot)$ is a mapping function, which maps the features into the RKHS. $\phi(\cdot)$ is associated with characteristic kernel k , $k(x^s, x^s) = \langle \phi(x^s), \phi(x^s) \rangle$. In theory [23], $p = q$ if and only if $\mathcal{L}_{\text{MMD}}(p, q) = 0$.

$\hat{\mathcal{L}}_{\text{MMD}}(p, q)$ is defined as an unbiased estimator of $\mathcal{L}_{\text{MMD}}(p, q)$, which can be calculated as follows:

$$\begin{aligned} \hat{\mathcal{L}}_{\text{MMD}}(p, q) &= \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(x_i^s) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(x_j^t) \right\|_{\mathcal{H}}^2 \\ &= \frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} k(x_i^s, x_j^s) + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} k(x_i^t, x_j^t) \\ &\quad - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} k(x_i^s, x_j^t). \end{aligned} \quad (2)$$

The LMMD in the proposed DASAN can be regarded as a weighted MMD, more details are shown in Section III.

III. PROPOSED METHOD

In this section, the architecture of DASAN is introduced in detail. The main idea of DASAN is to achieve global domain adaptation and subdomain adaptation simultaneously by learning the cross-domain invariant features.

A. Deep Adversarial Subdomain Adaptation Network

The proposed DASAN model consists of three modules: feature extractor, label classifier, and domain discriminator. The architecture of DASAN is shown in Fig. 2. A domain-shared 1-D CNN is designed as a feature extractor to extract high-level feature representations automatically from the raw data of the source domain and target domain data. Then, the extracted features are sent to a label classifier and a domain discriminator. The domain discriminator is trained to distinguish which domain the extracted features came from. The label classifier has three tasks. First, it is designed to predict the category labels of the extracted source domain or target domain features. Second, pseudolabel learning is newly introduced to suppress the uncertainty of the predicted label. Finally, the LMMD is introduced to measure the distribution discrepancy of relevant subdomains.

For the entire model, the feature extractor and classifier are trained using the labeled source domain to learn fault-special features and achieve subdomain adaptation by minimizing the LMMD. Additionally, pseudolabel learning minimizes the risk of the samples in the target domain being misclassified into irrelevant subdomains. Additionally, the minimax two-player game between the feature extractor and the domain discriminator is solved to align the global distributions of the source domain and target domain.

The detailed structure parameters of the feature extractor, label classifier, and domain discriminator are shown in Table I.

1) **Feature Extractor:** A feature extractor is designed to automatically extract high-level feature representations of the raw input data. It consists of three “convolution–pooling” modules, and each module contains one convolutional layer and one pooling layer. In the convolutional layer, the convolution kernels slide on the input data and perform convolution to extract features. Then, the rectified linear unit (ReLU) activation function is used to improve the nonlinearity of the model and avoid gradients vanishing and gradients exploding. Then, the extracted

TABLE I
STRUCTURE PARAMETERS OF DASAN

Networks	Layers	Operations
Feature extractor	Conv-Pool-1	Kernel $32 \times 5 \times 1$, Stride 1, Padding 2; BN; ReLU; Max-Pool 2×1 , Stride 2
	Conv-Pool-2	Kernel $32 \times 5 \times 1$, Stride 2, Padding 2; BN; ReLU; Max-Pool 2×1 , Stride 2
	Conv-Pool-3	Kernel $32 \times 5 \times 1$, Stride 2, Padding 2; BN; ReLU; Max-Pool 2×1 , Stride 2
	Flatten	Node 2048
Label classifier	Linear-1	Node 256; ReLU;
	Linear-2	Node: No. category; Softmax;
Domain classifier	Linear-1	Node 512; ReLU;
	Linear-2	Node 128; ReLU;
	Linear-3	Node 1; Sigmoid;

features are sent into a pooling layer (max-pooling) to perform a down-sampling operation to reduce the dimensions and retain feature information.

Specific to the model in this article, the input data are mapped to features f through mapping function G_f (feature extractor), and the mapping parameters are defined as θ_f , that is, $f = G_f(\cdot; \theta_f)$. And the feature extractor is shared across the source and target domains. The input data of the feature extractor are 1-D vibration signals with length L from the source domain and target domain, and the outputs of the feature extractor are D -dimensional features. Given the i th source sample x_i^s and j th target sample x_j^t as input data, the corresponding output features are z_i^s and z_j^t , respectively, where $z_i^s = G_f(x_i^s; \theta_f)$ and $z_j^t = G_f(x_j^t; \theta_f)$. In the training process, data are input into the model in a minibatch. The input data are x^s and x^t ($x^s, x^t \in \mathbb{R}^{M \times L}$), and the corresponding output features are $Z^s = G_f(x^s; \theta_f)$ and $Z^t = G_f(x^t; \theta_f)$ ($Z^s, Z^t \in \mathbb{R}^{M \times D}$), respectively, where M is the size of the minibatch.

2) Label Classifier: In domain adaptation, the label classifier is trained using labeled source domain samples instead of the unlabeled target domain samples. Minimizing the cross-entropy loss between the real label and predicted label guarantees the prediction accuracy of the label classifier. The label classifier can be applied in the target domain if the extracted features are invariant across domains. In the proposed DASAN, there are two fully connected layers in the label classifier. The ReLU activation function is used in the first layer, and the softmax activation function is used in the second layer. Detailed parameter information is shown in Table I. The extracted features are mapped to the predicted label \hat{y} by mapping function G_c (label classifier) with parameter θ_c , that is, $\hat{y} = G_c(Z^s; \theta_c)$. The loss of the label classifier is defined as follows:

$$\begin{aligned}
 \mathcal{L}_{\text{cls}} &= -\frac{1}{M} \left[\sum_{i=1}^M J_y(\hat{y}_i^s, y_i^s) \right] \\
 &= -\frac{1}{M} \left[\sum_{i=1}^M J_y(G_c(G_f(x_i^s; \theta_f); \theta_c), y_i^s) \right] \\
 &= -\frac{1}{M} \left[\sum_{i=1}^M \sum_{c=1}^{\mathbb{C}} I[y_i^s = c] \log(G_c(G_f(x_i^s; \theta_f); \theta_c)) \right]. \quad (3)
 \end{aligned}$$

3) Label Classifier: Domain adaptation urges the feature extractor to learn the domain-invariant features by minimizing the distribution discrepancy between domains. Existing intelligent

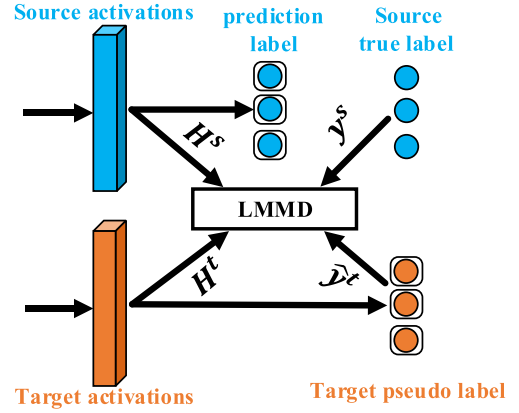


Fig. 3. LMMD module requires four inputs: source and target activations H^s, H^t , ($H^s, H^t \in \mathbb{R}^{M \times n}$) true label, y^s , and pseudolabel \hat{y}^t .

diagnosis methods based on moment matching or adversarial learning only align global domain distributions and lose fine-grained information. To solve this problem, the LMMD is introduced to consider the weights of samples according to the category to which the samples belong. The LMMD measures the discrepancy between the kernel-mean-embedding-related subdomains in the RKHS [24].

$$\mathcal{L}_{\text{LMMD}}(p^{(c)}, q^{(c)}) \triangleq E_c \|E_{p^{(c)}}[\phi(x^s)] - E_{q^{(c)}}[\phi(x^t)]\|_{\mathcal{H}}^2 \quad (4)$$

where x^s and x^t are samples from D_s and D_t , and $p^{(c)}$ and $q^{(c)}$ are distributions of subdomains $D_s^{(c)}$ and $D_t^{(c)}$, respectively. The parameter w^c is defined as the weight of each sample that belongs to each category. Hence, the unbiased estimation of (4) can be expressed as follows:

$$\hat{\mathcal{L}}_{\text{LMMD}}(p, q) = \frac{1}{\mathbb{C}} \sum_{c=1}^{\mathbb{C}} \left\| \sum_{x_i^s \in D^s} w_i^{Sc} \phi(x_i^s) - \sum_{x_j^t \in D^t} w_j^{Tc} \phi(x_j^t) \right\|_{\mathcal{H}}^2 \quad (5)$$

where w_i^{Sc} and w_j^{Tc} are the weights of x_i^s and x_j^t that belong to category c , respectively. In the minibatch, $\sum_{i=1}^M w_i^{Sc}$ and $\sum_{j=1}^M w_j^{Tc}$ are equal to 1. $\sum_{x_i \in D} w_i^c \phi(x_i)$ is the weighted sum of category c . For a given sample x_i , the weight is calculated as follows:

$$w_i^c = \frac{y_{ic}}{\sum_{(x_j, y_j) \in D} y_{jc}} \quad (6)$$

where y_{ic} is the label of x_i and the c th element of vector y_c . As shown in Fig. 3, for samples x_i^s from the labeled source domain, w_i^{Sc} is calculated using true label y_i^s . For samples x_j^t from the unlabeled target domain, w_j^{Tc} is calculated using the pseudolabel \hat{y}_j^t . In the proposed DASAN, the label classifier outputs the probability distribution of the target domain samples that reflect the probability of assigning x_j^t to each of the \mathbb{C} subdomains.

The LMMD is embedded into the first fully connected layer of the label classifier. In each batch, the activations of this layer are H^s and H^t ($H^s, H^t \in \mathbb{R}^{M \times n}$), where n is the number of neurons in the layer. The subdomain distribution discrepancy

is calculated as follows:

$$\begin{aligned} \hat{d}_l(p^{(c)}, q^{(c)}) = & \frac{1}{\mathbb{C}} \sum_{c=1}^{\mathbb{C}} \left[\sum_{i=1}^M \sum_{j=1}^M w_i^{Sc} w_j^{Sc} k(H_i^s, H_j^s) \right. \\ & + \sum_{i=1}^M \sum_{j=1}^M w_i^{Tc} w_j^{Tc} k(H_i^t, H_j^t) \\ & \left. - 2 \sum_{i=1}^M \sum_{j=1}^M w_i^{Sc} w_j^{Tc} k(H_i^s, H_j^t) \right] \cdot \quad (7) \end{aligned}$$

The related subdomain distributions can be aligned in different domains by minimizing $\hat{d}_l(p^{(c)}, q^{(c)})$. The labels of the target samples are required to calculate the LMMD, and the pseudolabels of the target domain samples can be obtained from the source classifier. However, the classifier is trained using supervised learning with labeled source samples. Pseudolabel learning [25] is introduced to solve the abovementioned problem. The pseudolabel learning loss is defined as follows:

$$\mathcal{L}_{\text{pseudo}} = -\frac{1}{M} \sum_{j=1}^M \sum_{m=1}^{\mathbb{C}} p(\hat{y}_j^t = m | x_j^t) \log p(\hat{y}_j^t = m | x_j^t) \quad (8)$$

where \hat{y}_j^t is the output of the classifier for the j th target sample.

The objective function of the label classifier can be written as follows:

$$\mathcal{L}_c = \mathcal{L}_{\text{cls}} + \lambda \hat{d}_l(p^{(c)}, q^{(c)}) + \gamma \mathcal{L}_{\text{pseudo}} \quad (9)$$

where λ and γ are tradeoff parameters.

4) Domain Discriminator: In the proposed DASAN, the domain discriminator module is designed to reduce the global distribution discrepancy and learn domain-invariant features. The domain discriminator consists of three fully connected layers, where the ReLU activation function is used in the first two layers and the sigmoid activation function in the last layer.

The extracted features are mapped to the domain labels d by mapping function G_d (domain discriminator) with corresponding parameters θ_d , that is, $d = G_d(f; \theta_d)$ ($x_i \in D_s$ if $d_i = 1$, $x_j \in D_t$ if $d_j = 0$).

The domain discriminator (G_d) plays an adversarial role with feature extractor (G_f), that is, they form a two-player minimax game. Specifically, the first player is the domain discriminator that is trained to distinguish the source domain and the target domain, and the second player is the feature extractor that is trained to deceive the domain discriminator [26]. The adversarial loss is defined as follows:

$$\begin{aligned} \mathcal{L}_{\text{adv}} = & -\frac{1}{M} \sum_{i=1}^M d_i \log[G_d(G_f(x_i^s; \theta_f); \theta_d)] \\ & - \frac{1}{M} \sum_{i=1}^M (1 - d_i) \log[1 - G_d(G_f(x_i^t; \theta_f); \theta_d)] \cdot \quad (10) \end{aligned}$$

B. Training Process

The objective function of the proposed DASAN model consists of the following four parts:

- 1) label classifier error;
- 2) LMMD of the subdomain adaptation;

TABLE II
TRAINING ALGORITHM FOR THE PROPOSED DASAN

Algorithm: training process for the proposed DASAN

1) Initialization

Input data $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, $D_t = \{(x_j^t)\}_{j=1}^{n_t}$. Initialize the parameters

$\theta_f, \theta_c, \theta_d$, features extractor G_f , label classifier G_c , and domain

discriminator G_d . Batch size is M . Learning rate is α .

2) Training

for each epoch do:

Forward propagation

random samples $x_s = \{(x_i^s)\}_{i=1}^M$, $x_t = \{(x_j^t)\}_{j=1}^M$

calculate $Z^s = G_f(x^s; \theta_f)$ and $Z^t = G_f(x^t; \theta_f)$, subdomain

features H^s and H^t , class label $\hat{y}^s = G_c(Z^s; \theta_c)$ and

$\hat{y}^t = G_c(Z^t; \theta_c)$, and domain label $G_d(Z^s; \theta_d)$ and $G_d(Z^t; \theta_d)$.

Back propagation

calculate the label classifier loss \mathcal{L}_{cls} using Eq. 3, LMMD loss

$\hat{\mathcal{L}}_{\text{LMMD}}$ using Eq. 7, pseudo-label learning loss $\mathcal{L}_{\text{pseudo}}$ using Eq. 8,

domain discriminator loss \mathcal{L}_{adv} using Eq. 10, and total loss $\mathcal{L}_{\text{total}}$ using Eq. 11.

Update the parameters $\theta_f, \theta_c, \theta_d$

$$\theta_f \leftarrow \theta_f - \alpha \left(\frac{\partial \mathcal{L}_{\text{cls}}}{\partial \theta_f} - \mu \frac{\partial \mathcal{L}_{\text{adv}}}{\partial \theta_f} + \lambda \frac{\partial \hat{\mathcal{L}}_{\text{LMMD}}}{\partial \theta_f} + \gamma \frac{\partial \mathcal{L}_{\text{pseudo}}}{\partial \theta_f} \right)$$

$$\theta_c \leftarrow \theta_c - \alpha \left(\frac{\partial \mathcal{L}_{\text{cls}}}{\partial \theta_c} + \lambda \frac{\partial \hat{\mathcal{L}}_{\text{LMMD}}}{\partial \theta_c} + \gamma \frac{\partial \mathcal{L}_{\text{pseudo}}}{\partial \theta_c} \right)$$

$$\theta_d \leftarrow \theta_d - \alpha \frac{\partial \mathcal{L}_{\text{adv}}}{\partial \theta_d}$$

end

3) Testing

Freeze all parameters of the DASAN model, and use the features extractor and label classifier to predict the target domain samples.

3) domain discriminator error;

4) pseudolabel learning constraint.

Combining (9) and (10), the objective function can be written as follows:

$$\mathcal{L}_{\text{total}}(\theta_f, \theta_c, \theta_d) = \mathcal{L}_{\text{cls}} - \mu \mathcal{L}_{\text{adv}} + \lambda \hat{\mathcal{L}}_{\text{LMMD}} + \gamma \mathcal{L}_{\text{pseudo}} \quad (11)$$

where λ , μ , and γ are the tradeoff parameters of the total loss.

According to [26], the optimization problem of adversarial learning is a minimax two-player game, that is, the loss of the domain discriminator is minimized to confuse the two domains and learn domain-invariant features by optimizing the parameter θ_f of the feature extractor, while simultaneously minimize the domain discriminator loss to distinguish the two domains by optimizing parameter θ_d of the domain discriminator. By contrast, the LMMD loss is minimized to align the subdomain distributions, and the label classifier loss is minimized to predict the label correctly. The training strategy of DASAN is shown in Table II, and the parameters θ_f, θ_c , and θ_d are optimized as

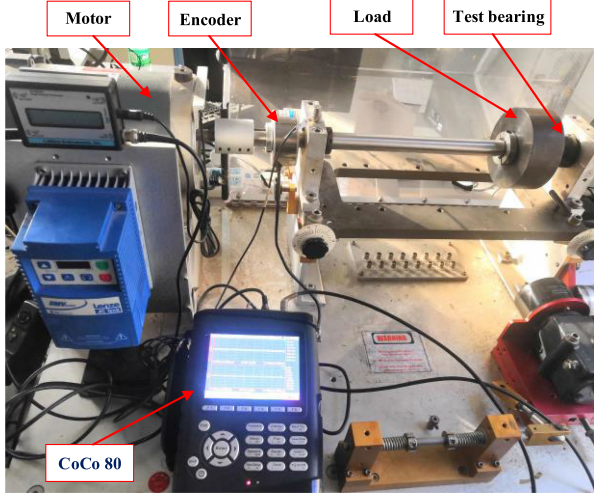


Fig. 4. Structure of the SQ test rig.

follows:

$$(\hat{\theta}_f, \hat{\theta}_c) = \arg \min_{\theta_f, \theta_c} \mathcal{L}_{\text{total}}(\theta_f, \theta_c, \hat{\theta}_d) \quad (12)$$

$$(\hat{\theta}_d) = \arg \max_{\theta_d} \mathcal{L}_{\text{total}}(\hat{\theta}_f, \hat{\theta}_c, \theta_d). \quad (13)$$

A gradient reversal layer (GRL) with parameter μ can be added between the feature extractor and the domain discriminator to solve the min-max optimization problem. GRL takes the gradient from the domain discriminator, multiplies it by $-\mu$, and passes it to the feature extractor; hence, μ is not optimized in the training process.

IV. EXPERIMENTAL RESULTS

A. Datasets

1) *Spectra Quest Dataset*: We collected the vibration data of the rolling bearings from the simulation test rig shown in Fig. 4, which was designed to simulate types of mechanical faults. The simulation test consisted of a load, test bearing, a motor, and a vibration sensor (COCO 80) for collecting data. There were four health conditions for each test bearing: normal condition, inner race fault, outer race fault, and ball fault. For the inner race fault and outer race fault, each condition had two severity levels (2 and 4 mm). The sampling frequency was 10 240, and the data were collected under three speeds (300, 480, and 660 r/min). Additionally, we regarded the data collected under different speeds as different domains for the transfer tasks. The proposed DASAN was evaluated on two transfer cases. The first case was bearing vibration signal data from the same machine but at different speeds. The second case was bearing vibration signal data from different rotating machines. The Spectra Quest (SQ) and CWRU datasets are described in Table III.

The first transfer case was based on the SQ dataset. We set four transfer tasks for SQ data: mutual transfer task between 300 and 480 r/min (300 r/min \Leftrightarrow 480 r/min) and mutual transfer task between 480 and 660 r/min (480 r/min \Leftrightarrow 600 r/min). There were five health conditions for each domain: normal, inner race fault (2 mm), inner race fault (4 mm), outer race fault (2 mm), and outer race fault (4 mm). We preprocessed raw vibration signals

TABLE III
DESCRIPTION OF DATASET

Method	SQ (Case 1)	SQ(Case 2)	CWRU
Sampling frequency	10240Hz	10240Hz	12000Hz
Sample length	2048	2048	2048
Health conditions	5	4	4
Samples for dataset	5*1000	4*1000	4*1000
Conditions	300/480/660 rpm	300/480/660 rpm	0HP (14)/2HP(7)/3HP(21)
States	NC/IF2/IF4/OF4/OF4	NC/IF/OF/BF	NC/IF/OF/BF

to construct the input data for the model. In the random sampling process, there was an 80% overlap between the current sample and the previous sample. As a result, the constructed training dataset consisted of five health conditions, and each condition had 1000 samples with a length of 2048.

2) *CWRU Bearing Dataset*: To accomplish the transfer task between different machines, we selected a public rolling bearing dataset CWRU [27], which was provided by Case Western Reserve University. The data were collected under four conditions: 1797 r/min/0 HP, 1772 r/in/1 HP, 1750 r/min/2 HP, and 1730 r/min/3HP. In this article, we used drive-end bearing data, which consisted of four health conditions: normal condition, inner race fault, outer race fault, and ball fault. There were three fault diameters for each condition: 7, 14, and 21 mils (1 mil = 0.001 inches). Additionally, the sampling frequency was 12 kHz.

We used the CWRU dataset for the second transfer case, and selected drive-end bearing data with 0 HP (14 mils), 2 HP (7 mils), and 3 HP (21 mils). For the SQ data, we selected data with a 4-mm fault size for the three conditions. We used the same preprocessing procedures for the CWRU dataset and SQ dataset. For the second transfer case, both the CWRU and SQ dataset consisted of four health conditions: normal condition, inner race fault, outer race fault, and ball fault. Additionally, each condition had 1000 samples with a length of 2048.

B. Transfer Results

1) *Comparison Methods*: To evaluate the proposed DASAN method comprehensively, the transfer results of DASAN were compared with those of five other methods.

- 1) The CNN consists of a feature extractor and a label classifier. The CNN model was trained using only labeled source domain data and applied in the target domain directly without domain adaptation.
- 2) Deep domain confusion (DDC) [29] utilizes the convolutional layers to extract common features and the fully connected layers to extract task-specific features. In the meantime, the MMD adaptation layer is involved to train the feature extractor to learn invariant features.
- 3) CORAL [28] is similar to DDC and uses the correlation alignment as second-order moment matching to reduce the distribution discrepancy. CORAL is embedded into the adaptation layer before the last fully connected layer.
- 4) The domain adversarial neural network (DANN) [30] is a basic architecture for adversarial-based domain adaptation. It consists of a feature extractor, a label classifier, and a domain discriminator. DANN uses GRL to solve

TABLE IV
DIAGNOSIS ACCURACY OF SQ DATA

Method	300 \Rightarrow 480	480 \Rightarrow 300	480 \Rightarrow 660	660 \Rightarrow 480	Average
CNN	0.492	0.561	0.443	0.354	0.463
DDC	0.892	0.868	0.841	0.867	0.867
CORAL	0.877	0.864	0.819	0.846	0.852
DANN	0.874	0.928	0.947	0.886	0.909
DCTLN	0.903	0.922	0.95	0.926	0.925
DASAN(w/o)	0.963	0.948	0.972	0.943	0.957
DASAN	0.976	0.962	0.983	0.967	0.972

the min-max optimization problem and learns invariant features by confusing the two domains.

- 5) DCTLN reduces the distribution discrepancy between the two domains using adversarial domain adaptation and minimizing the MMD loss [17].

The proposed DASAN is similar to DCTLN. However, DCTLN only focuses on global domain adaptation, whereas DASAN achieves global domain adaptation and subdomain adaptation simultaneously.

The abovementioned intelligent diagnosis methods based on domain adaptation use different network architectures. Thus, comparing the transfer results directly would not result in a convincing analysis. To compare them fairly, we designed the abovementioned methods to have the same basic architecture as DASAN, that is, the feature extractor, the label classifier, and the adversarial module had the same architecture parameters [31]. To demonstrate the effectiveness of pseudolabel learning, we added another comparison method called DASAN (w/o) to represent DASAN without pseudolabel learning.

For the proposed DASAN, the detailed parameters of the training process were set as follows: Raw data were used as the input for the network. To learn fault-special features for the label classifier and suppress the noisy influence of the domain discriminator in the early stages of training, we gradually changed the tradeoff parameters λ and μ , and set both of them to

$$\frac{2}{1 + \exp(-l \cdot s)} - 1 \quad (14)$$

where we set l to 10, gradually varied s from 0 to 1 in the training process [26], and selected γ from $\{0.001, 0.01, 0, 1\}$.

We chose the SGD optimizer with a momentum of 0.9 and weight decay of 5×10^{-4} to train the models. We set the dynamic learning rate α with an annealing strategy: $\alpha = 0.01/(1 + 10 * s)^{0.75}$. We set the batch size and training epochs to 32 and 200, respectively, for the abovementioned methods. We set the tradeoff parameters λ and μ in DDC, DANN, and DCTLN according to (14).

2) *Case 1: Results for Transfer Tasks for SQ Data:* For the domain adaptation problem in this article, we trained the diagnosis model with labeled source data and unlabeled target domain data. We used the classification accuracy to evaluate the performance of the trained model. We evaluated the methods on four transfer tasks under three rotating speeds. The experimental results of the proposed DASAN and five other domain adaptation methods are shown in Table IV.

According to the results shown in Table IV, the proposed DASAN method exhibited a 97.2% average accuracy, which was

TABLE V
COMPUTATION TIME FOR SQ DATA (TASK 480 \Rightarrow 660)

Method	CNN	DDC	CORAL	DANN	DCTLN	DASAN
Time (s)	891.1	1559.3	1487.3	1793.7	1912.9	2181.6

the best performance among all the methods. The CNN achieved the worst performance, with an average accuracy of 46.3%. The methods based on domain adaptation, including DDC, CORAL, DANN, and DCTLN, demonstrate their superior performance over the CNN method, which shows that domain adaptation is significant for intelligent transfer fault diagnosis. DASAN (w/o) achieved an accuracy of 95.7%, and DASAN achieved an accuracy of 97.2%, which demonstrates the effectiveness of the pseudolabel learning scheme.

Furthermore, DDC, CORAL, DANN, and DCTLN only aligned the global source and target distributions. However, the proposed DASAN focused on both global and local domain adaptations. Compared with the global domain adaptation methods, the local subdomain adaptation method used fine-grained information and the relationships between relevant subdomains to learn discriminative features. This proved to be effective in enhancing the classification performance of target samples. The abovementioned analysis demonstrates the superiority of the proposed DASAN.

The computational time of different methods for the transfer task 480 r/min \Rightarrow 600 r/min is shown in Table V. The proposed method could be implemented within 40 min. DASAN used the slightly more computational time to compute the weight of each target sample that belonged to each subdomain. However, increasing the computational time slightly was a worthwhile tradeoff for a significant improvement in performance, particularly for the transfer diagnosis model (trained offline).

Additionally, we used the t-distributed stochastic neighbor embedding (t-SNE) [32] method to analyze the transfer results and diagnosis performance intuitively. We chose the transfer task 480 r/min \Rightarrow 660 r/min, and selected 200 random samples for each health condition of different domains. We chose the outputs of the first fully connected layer of the label classifier as high-dimensional features and reduced them to two-dimensional features. The visualization results are shown in Fig. 5. For the result for raw data, all the health conditions overlapped substantially, which resulted in indistinguishable categories. Fig. 5(b) shows the visualization results from the CNN without domain adaptation. Different health conditions were well separated in the corresponding domains. However, the same health conditions from different domains exhibited significant distribution discrepancies. DANN and DCTLN reduced the distribution discrepancy through learning the invariant features across domains. However, we found that there was a clear boundary between the same categories from different domains. In Fig. 5(e), it is clear that DASAN achieved domain adaptation by aligning the global domain distribution and local subdomain distribution, which made the domain invariant features become better interclass discrimination and intraclass compactness.

3) *Case 2: Results for Transfer Tasks Between CWRU and SQ:* The robust intelligent diagnosis method should perform well on data collected from different but related machines. To demonstrate the generalization performance of DASAN, we evaluated it on six transfer tasks between CWRU and SQ. The

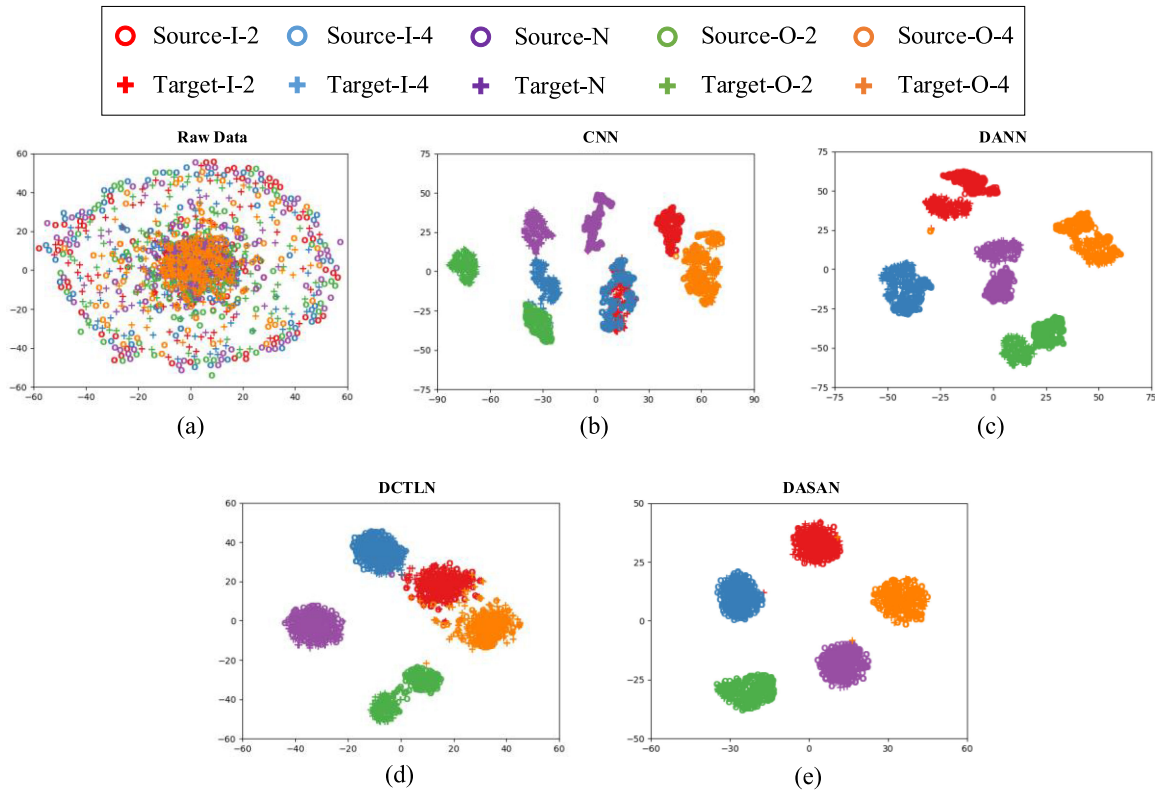


Fig. 5. Visual results for the raw data and four methods using *t*-SNE: (a) Raw data. (b) CNN. (c) DANN. (d) DCTLN. (e) DASAN.

TABLE VI
DIAGNOSIS ACCURACY OF THE TRANSFER TASKS ON DIFFERENT MACHINES

Method	300 \Rightarrow 0HP	0HP \Rightarrow 300	480 \Rightarrow 2HP	2HP \Rightarrow 480	660 \Rightarrow 3HP	3HP \Rightarrow 660	Average
CNN	0.428	0.504	0.520	0.495	0.751	0.745	0.574
DDC	0.645	0.796	0.841	0.897	0.918	0.893	0.832
CORAL	0.834	0.859	0.893	0.789	0.875	0.824	0.846
DANN	0.855	0.928	0.825	0.879	0.884	0.863	0.872
DCTLN	0.876	0.935	0.915	0.912	0.927	0.907	0.912
DASAN(w/o)	0.954	0.962	0.947	0.971	0.969	0.953	0.959
DASAN	0.977	0.988	0.967	0.984	0.982	0.983	0.980

transfer results are shown in Table VI. The proposed DASAN achieved the highest accuracy (98.0%). Although the other four transfer learning methods were superior to the CNN, their performance was still inferior to the proposed DASAN by over 6%. DANN, DDC, and CORAL had larger fluctuations across six tasks. The proposed DASAN had robust performance on transfer tasks on the same machine or different machines because of the inclusion of fine-grained information. The extensive experimental results demonstrate that the proposed DASAN method trained an effective diagnosis model using labeled data from one machine and transferred it to other different but related machines.

V. CONCLUSION

In this article, we proposed DASAN for intelligent fault diagnosis. The proposed method improved the learning mechanism and generalization performance significantly because of

the inclusion of the fine-grained information from the relevant category. Additionally, subdomain adaptation provided a novel perspective for cross-domain intelligent fault diagnosis. We verified DASAN on two datasets and extensive transfer tasks. The experimental results illustrated its transfer capability. We demonstrated the superiority of DASAN by comparing its transfer results with those of five other methods. Moreover, DASAN performed well on transfer tasks between different but related machines, that is, DASAN used acquired labeled data efficiently that contained rich health conditions and fault knowledge to diagnose other machines without labeled data. Finally, the proposed transfer learning methodology will be particularly useful in most of the actual industrial scenarios.

REFERENCES

- [1] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98–109, May 2017.

- [2] Y. Lei, Z. He, and Y. Zi, "A new approach to intelligent fault diagnosis of rotating machinery," *Expert Syst. Appl.*, vol. 35, no. 4, pp. 1593–1600, Nov. 2008.
- [3] Y. Lei, F. Jia, J. Lin, S. Xing, and S. X. Ding, "An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data," *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3137–3147, May 2016.
- [4] M. Zhao, J. Jiao, and J. Lin, "A data-driven monitoring scheme for rotating machinery via self-comparison approach," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2435–2445, Apr. 2019.
- [5] R. Zhang, H. Tao, L. Wu, and Y. Guan, "Transfer learning with neural networks for bearing fault diagnosis in changing working conditions," *IEEE Access*, vol. 5, pp. 14347–14357, Jun. 2017.
- [6] Y. Lei, N. Li, L. Guo, N. Li, T. Y an, 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.
- [7] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.
- [8] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
- [9] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 53–69, May 2015.
- [10] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Oct. 2018.
- [11] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Scholkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, Jul. 2006.
- [12] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschlager, and S. Saminger-Platz, "Central moment discrepancy (CMD) for domain-invariant representation learning," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–13. [Online]. Available: <https://arxiv.org/abs/1702.08811>
- [13] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 2058–2065.
- [14] L. Wen, L. Gao, and X. Li, "A new deep transfer learning based on sparse auto-encoder for fault diagnosis," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 49, no. 1, pp. 136–144, Jan. 2019.
- [15] 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.
- [16] T. Han, C. Liu, W. Yang, and D. Jiang, "A novel adversarial learning framework in deep convolutional neural network for intelligent diagnosis of mechanical faults," *Knowl.-Based Syst.*, vol. 165, pp. 474–487, Feb. 2019.
- [17] L. Guo, Y. Lei, S. Xing, T. Y an, and N. Li, "Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data," *IEEE Trans. Ind. Electron.*, vol. 66, no. 9, pp. 7316–7325, Sep. 2019.
- [18] Y. Wang, X. Sun, J. Li, and Y. Yang, "Intelligent fault diagnosis with deep adversarial domain adaptation," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021, doi: [10.1109/TIM.2020.3035385](https://doi.org/10.1109/TIM.2020.3035385).
- [19] 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.
- [20] X. Li, W. Zhang, Q. Ding, and J.-Q. Sun, "Multi-layer domain adaptation method for rolling bearing fault diagnosis," *Signal Process.*, vol. 157, pp. 180–197, Apr. 2019.
- [21] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Proc. 32nd AAAI Conf. Artif.*, 2018, pp. 1–8.
- [22] J. Wang, Y. Chen, H. Yu, M. Huang, and Q. Yang, "Easy transfer learning by exploiting intra-domain structures," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2019, pp. 1210–1215.
- [23] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, Mar. 2012.
- [24] Y. Zhu *et al.*, "Deep subdomain adaptation network for image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 4, pp. 1713–1722, Apr. 2021.
- [25] H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 2–8.
- [26] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by back-propagation," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.
- [27] W. A. Smith and R. B. Randall, "Rolling element bearing diagnostics using the case western reserve university data: A benchmark study," *Mech. Syst. Signal Process.*, vol. 64, pp. 100–131, Dec. 2015.
- [28] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 443–450.
- [29] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*.
- [30] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, May 2015.
- [31] Q. Wang, G. Michau, and O. Fink, "Domain adaptive transfer learning for fault diagnosis," in *Proc. Prognostics Syst. Health Manage. Conf. (PHM-Paris)*, 2019, pp. 279–285.
- [32] L. van der Maaten, "Accelerating t-SNE using tree-based algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3221–3245, Oct. 2014.



Yanxu Liu received the B.Eng. degree in mechanical engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2020. He is currently working toward the M.S. degree in mechanical engineering with Xi'an Jiaotong University, Xi'an, China.

His current research interests include fault diagnosis and prognostics, transfer learning, and health management.



Yu Wang (Senior Member, IEEE) received the B.Eng. degree in mechanical design and manufacturing automation from the Xi'an University of Technology, Xi'an, China, in 2005, the M.Eng. degree in manufacturing engineering and automation from Xi'an Jiaotong University, Xian, China, in 2008, and the Ph.D. degree in systems engineering and engineering management from the City University of Hong Kong, Hong Kong, in 2014.

Since 2017, he has been an Associate Professor with the School of Mechanical Engineering, Xi'an Jiaotong University. His current research interests include reliability assessment and fault prognostics and health management.



Tommy W. S. Chow (Fellow, IEEE) received the B.Sc. (First Class Hons.) and Ph.D. degrees from the Department of Electrical and Electronic Engineering, University of Sunderland, Sunderland, U.K., in 1984 and 1988, respectively.

He is currently a Professor with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. He has authored and coauthored more than 170 technical Journal articles related to his research, five book chapters, and one book. His main research interests include neural networks, machine learning, pattern recognition, and fault diagnosis.

Prof. Chow received the Best Paper Award in 2002 IEEE Industrial Electronics Society Annual meeting in Seville, Spain. He is currently an Associate Editor for IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS and *Neural Processing Letters*.

Prof. Chow received the Best Paper Award in 2002 IEEE Industrial Electronics Society Annual meeting in Seville, Spain. He is currently an Associate Editor for IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS and *Neural Processing Letters*.



Baotong Li received the bachelor's and master's degrees from Central South University, Changsha, China, in 2004 and 2007, respectively, and the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 2013 all in mechanical engineering.

He is currently a Professor with the School of Mechanical Engineering, Xi'an Jiaotong University. His research interests include mechanical design, optimization algorithm, and artificial intelligence science.