Adversarial Multiple-Target Domain Adaptation for Fault Classification

Article *in* IEEE Transactions on Instrumentation and Measurement · July 2020

DOI: 10.1109/TIM.2020.3009341

7 authors, including:

Mohamed Ragab
Nanyang Technological University
12 PUBLICATIONS **15** CITATIONS
SEE PROFILE

Zhenghua Chen
Institute for Infocomm Research
73 PUBLICATIONS **2,789** CITATIONS
SEE PROFILE

Min Wu
Institute for Infocomm Research
133 PUBLICATIONS **2,244** CITATIONS
SEE PROFILE

Haoliang Li
City University of Hong Kong
61 PUBLICATIONS **962** CITATIONS
SEE PROFILE

Adversarial Multiple-Target Domain Adaptation for Fault Diagnosis

Mohamed Ragab, Zhenghua Chen, Min Wu, Haoliang Li, Chee-Keong Kwoh, Ruqiang Yan, and Xiaoli Li

***Abstract*—Data-driven fault diagnosis methods are receiving great attention as they can be applied to many real-world applications. However, they work under the assumption that training data and testing data are drawn from the same distribution. Practical scenarios have varying operating conditions, which results in a domain shift problem that significantly deteriorate the diagnosis performance. Recently domain adaptation has been explored to address the domain shift problem by transferring the knowledge from labeled source domain (e.g., source working condition) to unlabeled target domain (e.g., target working condition). Yet all the existing methods are working under single source *single* target (1S1T) settings. Hence, a new model need to be trained for each new target domain. This shows limited scalability in handling multiple working conditions since different models should be trained for different target working conditions, which is clearly not a viable solution in practice. To address this problem, we propose a novel *adversarial multiple domain adaptation* (AMDA) method for single source *multiple* target (1SmT) scenario, where the model can generalize to multiple target domains concurrently. Adversarial adaptation is applied to transform the multiple target domains features to be invariant from the single source domain features. This leads to a scalable model with a novel capability of generalizing to multiple target domains. Extensive experiments on two public datasets and one self-collected dataset have demonstrated that the proposed method outperforms state-of-the-art methods consistently. Our source codes and data are available via https://github.com/mohamedr002/AMDA.**

***Index Terms*—Intelligent fault diagnosis, adversarial domain adaptation, discriminator, convolutional neural network, single source multiple targets**

I. Introduction

Data-driven fault diagnosis methods have potentials to generate great impacts in many real-world industrial applications. For example, it can help to intelligently monitor machine health status, identify root causes of failures, make maintenance decisions, etc. While traditional machine learning techniques have been employed for machine fault diagnosis [1], they suffer from labor-intensive feature engineering and require large amount of manually labelled training data.

During past few years, deep learning, with the ability to automatically extract salient features, achieves better performance in a few areas, including computer vision, speech recognition, natural language processing, etc. Recently, deep learning has also been applied for fault diagnosis. Chen *et al.* employed 1D convolutional neural network (CNN) with transferable features to leverage knowledge from source domain for fault diagnosis of rotary machinery [2], while Wen *et al.* developed a hierarchical diagnosis approach based on CNN to diagnose the fault and find its degradation level concurrently [3]. Sohaib *et al.* integrated CNN with bi-spectrum analysis to achieve fault diagnosis of inconsistent working environments [4]. In [5], stacked autoencoder was augmented with compressed sensing to reduce the amount of measured data and automatically extract features in a transform domain. Wang *et al.* integrated CNN with squeeze and excitation networks to graphically represent the bearing states [6]. Liang *et al.* employed semi-supervised generative adversarial network coupled with wavelet transform to reduce the number of labeled samples [7].

Zhao *et al.* performed a comprehensive review on different deep learning algorithms for fault diagnosis [8]. Nevertheless, these methods work under the assumption that labelled training data and unlabeled test data are drawn from the same distribution, which does not hold for many practical scenarios. For example, the training data could be collected under a certain working condition (e.g. 1 hp/horsepower working loads), and we can build models using existing methods, that often work well in test with the same working condition. However, in real-world applications, we may need to handle the real test data (unlabeled) with totally different working conditions (e.g., 2 hp or any other working loads), meaning that the distribution

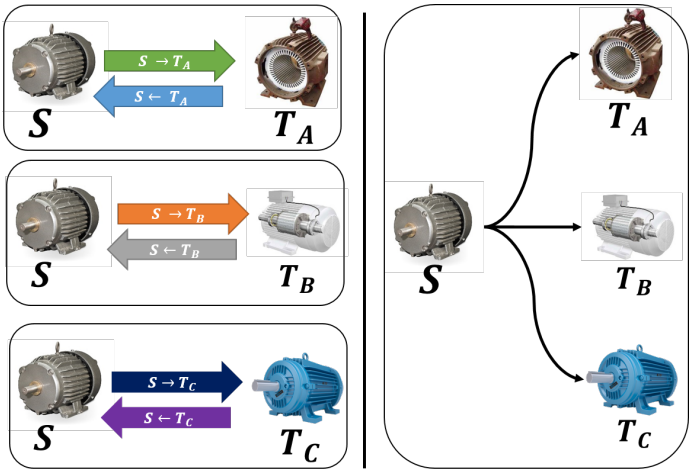

Fig. 1. Existing approaches vs our scalable multi-target approach

Mohamed Ragab and Chee-Keong Kwoh are with school of Computer Science and Engineering at Nanyang Technological University, Singapore (Email: mohamedr002@e.ntu.edu.sg, asckkwoh@ntu.edu.sg).

Zhenghua Chen, Min Wu, and Xiaoli Li are with Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), 1 Fusionopolis Way, Sinagpore 138632 (Email: chen0832@e.ntu.edu.sg, wumin@i2r.a-star.edu.sg, xlli@i2r.a-star.edu.sg).

Haoliang Li is with School of Electrical and Electronic and Engineering at Nanyang Technological University, Singapore (Email: hli016@e.ntu.edu.sg).

Ruqian Yan is with School of Mechanical Engineering, Xi'an Jiaotong University, China (Email: yanruqiang@xjtu.edu.cn).

of the unlabeled test data *usually* do not follow the same distribution as the labeled training data. Thus the trained classifier will not be able to generalize well on test data with different distributions. As such, we need to recollect a set of training data to rebuild a customised model, specifically for each working condition. However, it is very expensive, if not impossible, to annotate training data for each working condition to rebuild a new model.

Recently, domain adaptation (DA), a special case of transfer learning, has been proposed to leverage the knowledge from labelled source domain data to train a classifier that can generalize to a target domain with a different distribution. DA has been successfully applied in many different applications such as natural language processing, object recognition, speech recognition, and sentiment analysis [9]. Very recently, it has been explored to address domain shift problem to transfer the model from source domain (one working condition) to target domain (different working condition) in intelligent fault diagnosis and prognosis problems [10]–[16]. However, all existing methods work under single source *single* target settings (1S1T), which *is not feasible as the working conditions can be varying* to satisfy different manufacturing needs. As such, if the target domain has changed, we need to train a new model independently as shown in Fig. 1, which is clearly not a viable solution in practice. On the other hand, naïvely merging multiple target domains together into a single target will not work either, as data from multiple target domains typically have different data distributions and unique data characteristics.

In this paper, we build upon the work done by *Tzeng et al.*[17], who proposed adversarial domain adaptation approach with (1S1T) to obtain domain invariant features for image-related problems. We extend this work in two directions. First, we realize the adversarial domain approach for time series data. Second, we tackle a more challenging and practical domain adaptation problem under the single source and multiple targets (1SmT) setting for fault diagnosis purpose. For instance, we assume that a machine can work under four different loads, i.e., A, B, C and D. Some data have been collected to train a fault diagnostic model when the machine is working under load A. In our 1SmT setting, the model can adapt to multiple different loads concurrently, i.e., B, C and D. We propose a novel deep learning architecture for adversarial unsupervised domain adaptation for the 1SmT problem. As shown in Fig. 2, we first train the source feature extractor to obtain class discriminative features using the labeled source domain. Then, the target feature extractors are initialized by the weights of the source feature extractor and thus inherit the class-discriminative property. On the other hand, a discriminator network is trained to distinguish between the source and multiple targets features. To obtain domain invariant features among different targets, we adversarially update multiple target feature extractors to generate features that can be indistinguishable for the discriminator. During testing, our scalable model can take any of the target domains and generate source-like features, where the trained source classifier is able to generalize well to any of the targets.

The main contributions of this paper can be summarized as follows.

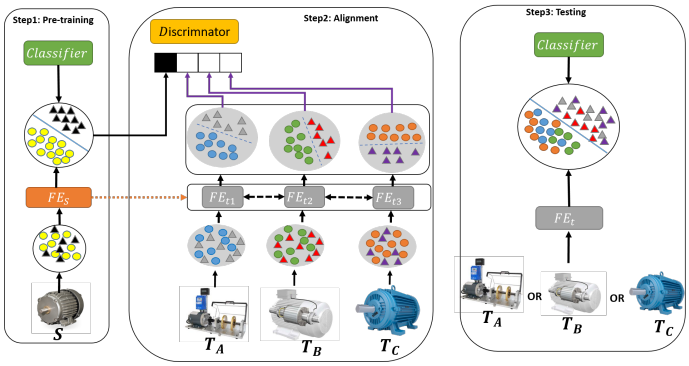

Fig. 2. Proposed adversarial multiple target domain adaptation for fault diagnosis.

- We formulate a more realistic 1SmT problem that is particularly used for real-world fault diagnostic problem.
- We propose a novel multiple adversarial domain adaptation (AMDA) method, which designs a deep learning architecture for adversarial unsupervised domain adaptation to address 1SmT problem. To the best of our knowledge, it is the first attempt in this area.
- We addressed the limited scalability of existing approaches by proposing a general model that can generalize to multiple targets concurrently.
- Extensive experimental results demonstrate that our proposed AMDA model can generalize to multiple target domains simultaneously and achieve significantly better results than the state-of-the-art methods consistently.

II. Related Works

Unsupervised domain adaptation (DA) transfers knowledge to source domain with sufficient labels to unlabeled target domain data drawn from a different but related distribution. In the fault diagnosis problem, many approaches have been developed to address the domain shift problem. However, they only work with single source single target (1S1T) scenario, which can only handle single target domain at a time. Differently, we propose a novel single source multiple targets (1SmT) scenario to handle multiple targets concurrently, which is more scalable and valuable for practical fault diagnosis problems.

A. Single Source Single Target

Many existing approaches have employed domain adaptation for fault diagnosis using single source single target scenario (1S1T) [18]. In [19], researchers employed auto-encoder to extract domain invariant features, with the help of popular domain discrepancy metric *Maximum Mean Discrepancy* (MMD) [20] to measure the discrepancy between the source and target distributions. Minimizing both auto-encoder loss and MMD loss between the two distributions will produce a good feature representation for both source and target domains. A wide kernel CNN with adapted batch normalization to improve the generalization was proposed by Zhang *et al.* [21]. Wen *et al.* [22] proposed sparse auto-encoder to extract features in an unsupervised manner that

is subsequently used to train a softmax classifier. Moreover, they also fine-tuned the network to minimize the classification loss and the domain discrepancy loss. Very recently, Li *et al.* [15] employed 1D CNN to extract feature representation from frequency domain features. They also used a representation clustering scheme to maximize intra-class similarity and reduce inter-class similarity, coupled with classification loss for more discriminative features, and adopted MMD to obtain domain invariant features [23]. In [10], Xiang *et al.* addressed the problem of cross domain fault diagnosis with insufficient faulty data, where they employed generative adversarial networks to generate faulty data in the target domain [24], and used the generated data into the domain adaptation scheme to solve the cross domain problem. In [25], a sparse filtering approach coupled with high-order Kullback-Leiblier divergence extracted a domain invariant features in unsupervised manner. Li *et al.* proposed a domain adaptation approach to address fault diagnosis problem with data from different places in the same machine [26]. Particularly, they integrated a gradient reversal layer with a novel parallel data alignment technique to tackle the domain shift problem. In [27]. a hierarchical deep domain adaptation approach has been used for fault diagnosis of thermal system under varying working conditions. Specifically, they employed correlation alignment (CORAL) with successive denoising autoencoders to learn domain invariant features among different working conditions. Finally, in [28], a two phase approach was proposed, where the authors first pre-trained a model on the source domain data using 1D CNN, and then fine-tuned untied model using target domain data and MMD.

B. Single Source Multiple Targets

Among domain adaptation literature, a little attention has been paid to (1SmT) problem. Recently, some approaches have addressed multiple domain learning problem [29], [30]. However, they all in the context of image generation task, where they train a single generator to generate samples from different domains. Differently, our AMDA approach is addressing (1SmT) for time series classification problem. To the best of our knowledge, our proposed AMDA is first trial in this application.

III. Adversarial Multiple-Target Domain Adaptation

In this section, we firstly present our problem formulation for single source multiple targets (1SmT), and then provide technical details on addressing the 1SmT problem with an application to time series data for fault diagnosis. The proposed framework as shown in Fig. 3 is composed of three main architectures, namely, Feature Extractor E, Classifier C, and Discriminator D. Specifically, we used E to construct single source feature extractor E_s, and multiple target feature extractors $E_{t(N)}$ with tied weights.

Different from existing approaches, we tie the weights of the feature extractors of the multiple target domains, inspired by multi-task learning [31]. This enables a single feature extractor to generalize to multiple target domains during testing stage. In addition, it helps to reduce the capacity of the model and act as a regularizer to avoid overfitting. To this end, unlike all existing approaches, which can generalize to single target at a time, our model can be more scalable and have a generalization ability that can handle multiple targets concurrently.

In general, our proposed method contains three main steps: (1) supervised learning using source domain labels; (2) adversarial adaptation of N target domains to single source domain; (3) test the domain adapted model on all N target domains. The goal of this paper is to construct a network that can find a shared latent space between the source and multiple target domains, such that the discrepancy between the source domain and target domains is minimized. As such, the model can be better generalized to the multiple target domains concurrently. In the following subsections, we will explain each step in more details.

A. Problem Formulation

The domain adaptation involves a domain $\mathcal{D}$ and task $\mathcal{T}$ [32], where the domain $\mathcal{D}$ consists of two components: a feature space $\mathcal{X}$ and marginal distribution $P(\mathbf{x})$, where $\mathcal{D} = \{\mathcal{X}, P(\mathbf{x})\}$, $\mathbf{x} \in \mathcal{X}$, where $\mathbf{x}$ is the data sample. Correspondingly, the task $\mathcal{T}$ consists of two components: a label space $\mathcal{Y}$ and mapping function $f(\mathbf{x})$, where $\mathcal{T} = \{\mathcal{Y}, f(\mathbf{x})\}$.

Our 1SmT problem can be formulated as follows:

- We have a *labelled* single source domain $\mathcal{D}_s = \{\mathbf{x}_s^i, y_s^i\}_{i=1}^{n_s}$ of n_s samples, where $\mathbf{x}_s^i \in \mathcal{X}_s$ is the data sample and $y_s^i \in \mathcal{Y}_s$ is the corresponding label. Similarly, we have *unlabeled* multiple target domains $\{\mathcal{D}_{t(1)}, \ldots, \mathcal{D}_{t(N)}\}$, where N is the number of target domains and $\mathcal{D}_{t(j)} = \{\mathbf{x}_{t(j)}^i\}_{i=1}^{n_t}$ represents the total samples of domain j. More specifically, $\mathbf{x}_{t(j)}^i \in \mathcal{X}_{t(j)}$ is the i^{th} sample of the target domain j, where $\mathcal{X}_{t(j)}$ is feature space and n_t is the number of unlabeled samples for the corresponding target domain.
- The feature space of the single source and multiple target domains is same, i.e., $\mathcal{X}_s = \mathcal{X}_{t(1)} = \mathcal{X}_{t(2)} = ... = \mathcal{X}_{t(N)}$, where N is the number of target domains.
- The marginal distribution between the source domain and target domains is different due to variation on multiple target domains (e.g. with different working conditions), i.e., $P_s(\mathbf{x}) \neq P_{t(j)}(\mathbf{x})$ $(j = 1, 2, ..., N)$. In addition, marginal distributions among different target domains are also different, i.e., $P_{t(j)}(\mathbf{x}) \neq P_{t(k)}(\mathbf{x})$, where $j \neq k$.
- Label space of the single source domain and multiple target domains is the same, i.e. $\mathcal{Y}_s = \mathcal{Y}_{t(1)} = \mathcal{Y}_{t(2)} = ... = \mathcal{Y}_{t(N)}$

B. Supervised Learning with Labelled Source Domain Data

Our first step employs the labelled source domain data $\mathcal{D}_s = \{\mathbf{x}_s^i, y_s^i\}_{i=1}^{n_s}$, where $y_s^i \in \{1, ..., k\}$ and k is the number of classes, to learn a feature extractor E_s and classifier C in supervised learning manner by minimizing the cross entropy loss between the predicted labels and ground-truth labels which is shown in the following equation.

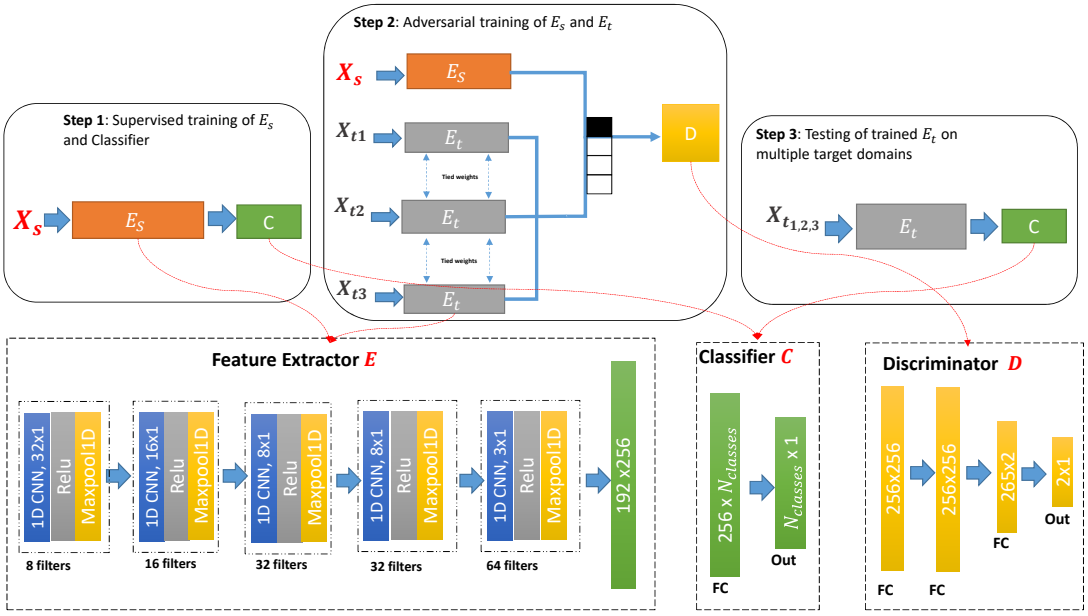

Fig. 3. Adversarial single source multiple target domain adaptation (AMDA) model for fault diagnosis with three main architectures: feature extractors (e.g., E_s for source domain and E_t for target domains), classifier C, and discriminator D.

$$L_{ce} = -\frac{1}{n_s}\sum_{i=1}^{n_s}\mathbf{1}\{y_s^i = j\}\log C(E_s(\mathbf{x}_s^i)) \tag{1}$$

Where L_{ce} is the cross entropy loss, $y_i \in \mathcal{Y}_s$, and $\mathbf{1}$ is an indicator function that return 1 when argument is true.

The parameters of feature extractor will be used in the next step for two purposes: 1) initialize the target domain feature extractors $E_{t(N)}$ to be inherently class discriminative, and 2) be used as a reference model during adversarial training. Algorithm 1 provides the pseudo code, including the details of training source feature extractor E_s under the supervision of source domain labels, by employing $\mathcal{D}_s$ to learn the parameters of E_s that can minimize the classification loss in Eq. 1.

Algorithm 1: Supervised Learning Using Labelled Data from Source Domain

Input: Single source domain: $\mathcal{D}_s = \{\mathbf{x}_s^i, y_s^i\}_{i=1}^{n_s}$, and batch size is m
Output: Trained source feature extractor E_s and classifier C
$E_s \leftarrow$ Convolutional neural network
$C \leftarrow$ Fully connected neural network
for *number of samples* **do**
1. $\mathbf{X}_s \leftarrow \{\mathbf{x}_s^1, \ldots, \mathbf{x}_s^m\}$, mini-batch of source samples
2. $Y_s \leftarrow \{y_s^1, \ldots, y_s^m\}$, mini-batch of source labels
3. $Preds \leftarrow C(E_s(\mathbf{X}_s))$
4. Train E_s and C using Eq. 1
5. Update the weights using *Adam* optimizer

end

C. Adversarial Training of Multiple Target Feature Extractors

The key idea of adversarial training is based on min-max game between the target feature extractor and the domain discriminator. More specifically, the discriminator network is trained to distinguish between the source and target features, while the target feature extractor is trained to maximize the discriminator loss by producing target features that is invariant from the source domain features [24]. Hence, the classifier trained on the source domain features can generalize well on the target domain features. Nevertheless, this approach can generalize well to only single domain at a time and for any change in the target or in the source domain you need to train new model independently. As such, to handle k working conditions you need to train k different models which is not a viable solution. In our work, we propose a scalable model that can handle multiple working conditions concurrently. We find a new shared feature representation among the multiple target domains that can be invariant from the source domain features in one training phase. Thus, the trained source classifier can generalize to the domain invariant features of the target domains. To do so, we tie the weights of all the target feature extractors during training phase. As a result, we can use the common weights of target feature extractors to map any of target domains to be invariant from the source domain features. In this section, we provide the detailed training process of our proposed approach

Our key idea is to provide an efficient framework to handle N target domains in one training phase, by training a discriminator against N target feature extractors simultaneously. Particularly, we pass $\{\mathbf{X}_{t(1)}, \ldots, \mathbf{X}_{t(N)}\}$ to N feature extractors with *tied weights* to produce $\{\mathbf{h}_{t(1)}, \ldots, \mathbf{h}_{t(N)}\}$. Then the discriminator network D will perform domain classification between the source domain features h_s and the target domain features. However, initially, the target domains features (e.g., $\{\mathbf{h}_{t(1)}, \ldots, \mathbf{h}_{t(N)}\}$) are very distinguishable from source domain features (e.g., h_s). Thus the discriminator loss can vanish and limit the domain alignment process. To prevent the resulted gradient vanishing, the discriminator is trained

every N iterations of training target feature extractors. Hence, the discriminator can push the N target feature extractors to map all the target domains to shared latent space, where the discrepancy between the source domain and these N target domains is minimized. The discriminator and multiple target feature extractors are trained with generative adversarial networks (GAN) loss [24]. In particular, the discriminator is trained using logistic function by assigning 1 to the source domain data, and 0 to the data in N target domains. The discriminator classifies each input sample and decide whether it belongs to the source domain or the target domains, under standard supervised learning fashion, where the loss is denoted as $\mathcal{L}_D$.

$$\min_D \mathcal{L}_D = -\mathbb{E}_{\mathbf{x}_s \sim P_s}\big[\log D(E_s(\mathbf{x}_s))\big] - \sum_{j=1}^{N} \mathbb{E}_{\mathbf{x}_{t(j)} \sim P_{t(j)}}\big[\log(1 - D(E_{t(j)}(\mathbf{x}_{t(j)})))\big]. \tag{2}$$

Where $\mathbf{x}_s$ is source domain sample , $\mathbf{x}_{t(j)}$ are the target domains samples with $(1 \leq j \leq N)$.

The objective function of the target feature extractors is defined as follows:

$$\min_{E_{t(1)},\dots E_{t(N)}} \mathcal{L}_E = -\sum_{j=1}^{N} \mathbb{E}_{\mathbf{x}_{t(j)} \sim P_{t(j)}}\big[\log D(E_{t(j)}(\mathbf{x}_{t(j)}))\big]. \tag{3}$$

where $E_{t(i)}$ is the feature extractor for the i^{th} target domain $(1 \leq i \leq N)$. By minimizing the loss function $\mathcal{L}_E$, the target feature extractors will map the target domain features to a shared latent space where the discrepancy between the centroid of all target distributions and source domain distribution is minimized.

Detailed steps for fine tuning phase are presented in Algorithm 2, where the parameters of $E_{t(N)}$ are derived such that the output features are domain invariant and class discriminative. Adversarial training is employed between N target feature extractors with tied layers and discriminator D to minimize $\mathcal{L}_D$ and $\mathcal{L}_E$.

D. Testing on the Target Domain

To justify our contribution by formulating the DA problem as 1SmT, we test the trained E_t to samples from any of N target domains, and then pass the output features to the pretrained classifier C to predict the class of the corresponding sample. Eq. 4 shows the usage of softmax to compute the probability of each class given the input instance from any target domains:

$$p(y_i = k|C) = \frac{\exp(C_k(\mathbf{F}_t))}{\sum_{k'} \exp(C_{k'}(\mathbf{F}_t))} \tag{4}$$

where F_t is latent representation of the corresponding target domain, $C_{k'}(\cdot)$ denotes the output of k_{th} class resulted from softmax.

Algorithm 2: Adversarial Training for Multiple Targets

Input: Single source domain : $\mathcal{D}_s = \{\mathbf{x}_s^i, y_s^i\}_{i=1}^{n_s}$, Multiple target domains: $\{\mathcal{D}_{t(1)}, \dots, \mathcal{D}_{t(N)}\}$, where with $\mathcal{D}_{t(j)} = \{\mathbf{x}_t^j\}_{i=1}^{n_t}$, N is number of target domains, and m is the batch size.
Output: Trained multiple target feature extractors $E_{t(1)}, \dots, E_{t(N)}$
$E_s \leftarrow$ Pretrained source feature extractor
$E_{t(N)} \leftarrow$ Initialize with source parameters E_s
$D \leftarrow$ Discriminator network
for *number of iterations* **do**
  1. Sample mini-batch of m source samples $\mathbf{X}_s \sim P_s$
  2. Sample mini-batch of m from each target domain: $\{\mathbf{X}_{t(1)}, \dots, \mathbf{X}_{t(N)}\} \sim \{P_{t(1)}, \dots P_{t(N)}\}$
  3. Extract source domain features: $E_s(\mathbf{X}_s)$
  4. Extract features from N target domains concurrently: $\{E_{t(1)}(\mathbf{X}_{t(1)}), \dots, E_{t(N)}(\mathbf{X}_{t(N)})\}$
  5. Update D by Eq. 2 `// Train Discriminator`
  for *M steps* **do** `// Train` E_t `M times`
    6. Extract features from N target domains: $\{E_{t(1)}(\mathbf{X}_{t(1)}), \dots, E_{t(N)}(\mathbf{X}_{t(N)})\}$
    7. Update the target feature extractor E_t by Eq. 3
  end
end

IV. Experiments

In this section, we evaluate the performance of our proposed AMDA model on fault diagnosis that needs to classify machine bearing health status into either normal or different classes of faults.

A. Implementation Details

In our model, we employed a five layer 1D CNN as a feature extractor and used wide input kernel for longer dependencies. Fully connected neural network with softmax layer was used for fault classification, while two layer fully connected network was used to discriminate between the source domain and target domain data. Fig. 3 illustrates the detailed implementation of both feature extractor and classifier. The learning rate of feature extractor and discriminator is set to be $1e$-4, which is small enough to avoid overshooting valley or minimum in the error surface, and thus yields the maximum generalization accuracy.

B. Case 1: Case Western Reserve University Dataset

1) Dataset Description: We have employed Case Western Reserve University (CWRU) [33] benchmark dataset, which has been collected from drive end of motor under 12k sampling rate. The data consists of four different subsets. Particularly, each subset represents a specific working condition, i.e., a specific working load from 0 to 3hp. Each subset has 4 different class labels for faults, i.e., normal and three types of faults, namely inner-race (IF), bearing-race (BF), and outer-race (OF) at centered position of @6:00 relative to the load zone. Moreover, each type of fault could have 3 different fault sizes, i.e., 0.007 inches, 0.014 inches, and 0.021 inches, which leads to 10 different classes (1 normal class, and 9

fault classes), as shown in Table I. In addition, we used sliding windows with overlaps on time series data for data augmentation to increase the number of samples [34]. The corresponding window width and shifting step is 4096 and 295 respectively. Eventually, each working condition has 4000 samples and each sample is represented as a 4096 dimensional vector.

TABLE I
CWRU BEARING DATASET DESCRIPTION [35]

Working Condition	Loading Torque	Fault Type	Fault Size (inches)
A	0 hp	Normal, IF, OF, BF	0, 007,0.014, 0.021
B	1 hp	Normal, IF, OF, BF	0, 007,0.014, 0.021
C	2 hp	Normal, IF, OF, BF	0, 007,0.014, 0.021
D	3 hp	Normal, IF, OF, BF	0, 007,0.014, 0.021

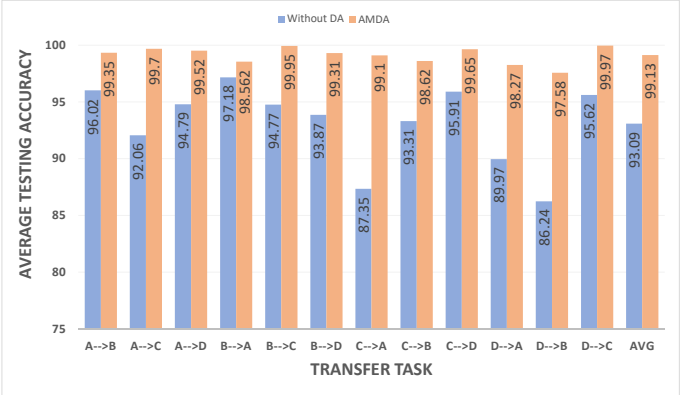

Fig. 4. Evaluation of AMDA with and without domain adaptation on CWRU dataset using 12 cross-domain scenarios

2) Experimental Results: We denote four working conditions as A, B, C and D, which correspond to load 0, 1, 2, and 3 respectively. To comprehensively evaluate the performance of our proposed AMDA model, we conducted 12 cross-domain experiments as shown in Fig. 4. For the first 3 experiments (A→B, A→C, A→D), we used working condition A as source domain, and B, C and D as multiple target domains to learn the feature extractors, classifier and discriminator. Then, we tested the learned feature extractor on each individual target domain B, C and D to generate the results for A→B, A→C and A→D, respectively. Similarly, we also used B, C, and D as our source domains for cross-domain experiments.

Fig. 4 shows the performance of our proposed AMDA model over 12 cross-domain experiments. Note that without DA in Fig. 4 refers to our AMDA model without the discriminator, i.e., directly using the source feature extractor for the target domain. Overall, our AMDA achieves an average accuracy of 99.13% over 12 experiments, which is 6.04% higher than without DA. These results demonstrate the effectiveness of domain adaptation in our model for cross-domain fault diagnosis. Note that, we use a 1-layer classifier C (see Fig. 3) in this work. Our empirical test demonstrates that if we use more layers for the classifier C, the AMDA without DA will perform even worse (i.e., the gap between AMDA with and without DA becomes larger) due to the general issue of overfitting.

In addition, there are some *easy* transfer cases, such as A→B and B→A scenarios, for which without DA can achieve an accuracy of 96.02% and 97.18% as shown in Fig. 4. Meanwhile, D→A and D→B scenarios are *hard* transfer cases, with performance of 89.97% and 86.24% respectively. With our proposed AMDA model, we can achieve improvement for both *easy* and *hard* transfer cases, e.g., 3.33% for A→B and 11.34% for D→B. Hence, AMDA can play a more important role and achieve better performance when domain discrepancies become larger and harder to transfer.

3) Comparison to Domain adaptation baselines: To demonstrate the superiority of the proposed AMDA, we implemented four domain adaptation baselines: Transfer Component Analysis (TCA) [36], Joint Distribution Adaptation (JDA) [37], Correlation Alignment (CORAL) [38], Deep Domain Confusion (DDC) [39], Deep Maximum Mean Discrepancy (MMD) [40], and Deep CORAL [41].

Table II shows the results of different domain adaptation techniques using CWRU dataset. It can be found that the DDC achieves the best performance among baselines with an overall accuracy of 96.25%. The proposed AMDA outperforms all the baseline techniques on 12 domain adaptation scenarioes with an overall accuracy of 99.13%, which indicates the effectiveness of the proposed AMDA for this domain adaptation task.

4) Comparison to State-of-the-arts: To better evaluate the performance of our proposed AMDA model, we have also conducted experiments to compare it with 3 different state-of-the-art baselines, which are summarized as follows.

- The first approach is fault diagnosis using deep neural network (DNN) [42], which consists of pre-training the stacked-autoencoder in unsupervised manner and fine-tuning the network under the supervision of source labels.
- The second approach is a 5-layer CNN with wide input kernel which was demonstrated to achieve high accuracy (WDCNN) [34].
- The last method is transfer inference with convolutional neural network (TICNN) with a 6-layer CNN and introduces dropout in the first input layer. Additionally ensemble learning has been employed to stabilize the performance of their model [43].

Table III shows the performance comparison between the proposed AMDA model with three state-of-the-art methods. For these three competing methods, they only reported their results on 6 cross-domain experiments. Therefore, we also conducted the same cross-domain experiments for fair evaluation.

We observe that our proposed AMDA method achieves better results than three existing methods consistently. Note that almost all the methods have achieved good results for *easy* transfer cases (e.g., A→B), however, they fail to achieve good results in more challenging tasks with high domain discrepancies (e.g., C→A). Nevertheless, with well-designed adversarial domain adaptation, our AMDA model is able to achieve significant improvements over all the state-of-the-art methods. Furthermore, this excellent performance is achieved under the challenging settings of 1SmT by adapting multiple

TABLE II
EVALUATION OF AMDA ON CWRU DATASET AGAINST DOMAIN ADAPTATION BASELINES USING 12 CROSS-DOMAIN SCENARIOS

	Method	A→B	A→C	A→D	B→A	B→C	B→D	C→A	C→B	C→D	D→A	D→B	D→C	AVG
Shallow	CORAL	53.73	49.29	49.21	79.74	74.72	78.76	71.41	62.55	62.19	75.48	73.17	68.25	66.55
	TCA	64.06	64.4	76.94	66.94	75.92	82.96	56.06	67.34	30.4	74.86	44.79	70.05	64.56
	JDA	71.35	66.25	82.23	67.69	73.68	83.76	54.49	66.10	60.32	75.86	80.25	70.61	71.05
Deep	DDC	95.62	98.42	95.04	95.56	98.33	99.06	95.83	97.17	97.29	86.42	96.62	99.62	96.25
	Deep MMD	97.27	90.60	94.69	96.23	98.88	97.90	94.60	96.63	93.6	95.25	95.50	99.06	95.85
	Deep CORAL	88.73	87.13	97.52	97.58	98.75	98.38	94.54	96.04	97.21	96.10	96.52	98.19	95.56
	AMDA	**99.35**	**99.70**	**99.52**	**98.56**	**99.95**	**99.31**	**99.10**	**98.62**	**99.65**	**98.27**	**97.58**	**99.97**	**99.13**

TABLE III
COMPARISON WITH RELATED WORKS ON 6 TRANSFER SCENARIOS

Method	A→B	A→C	B→A	B→C	C→A	C→B	AVG
DNN	82.2	92.6	72.3	77.0	76.9	77.0	79.60
WDCNN	99.2	91.0	95.1	91.5	78.1	85.1	90.00
TICNN	99.1	90.7	97.4	98.8	89.2	97.6	95.47
AMDA	**99.4**	**99.7**	**98.6**	**99.9**	**99.1**	**98.6**	**99.21**

targets simultaneously in one training phase, in comparison with only one single target at a time for all the competing methods.

C. Case 2: KAt Bearing Dataset

1) Dataset Description: KAt bearing dataset was collected using the modular rig tester as shown in Fig. 5 [44]. The tester consists of several components: (1) electric motor, (2) torque-measurement shaft, (3) a rolling bearing test module, (4) fly wheel and (5) load motor. More details about the modular tester for data collection can be found in [44]. In this dataset, 32 experiments for rolling bearing elements were conducted to collect 3 types of data, namely, undamaged bearing data, artificially damaged bearing data and real damaged bearing data. In particular, the bearing data in each experiment has 20 files and each file was collected for 4 seconds with a sampling rate of 64 KHz.

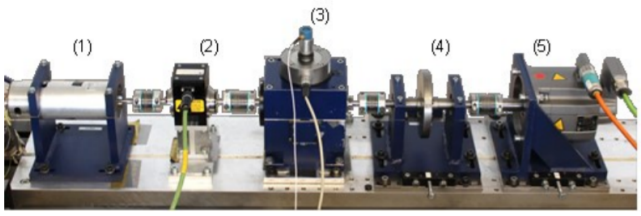

Fig. 5. Modular test rig for collecting the KAt dataset [44]

To generate the data samples, we also used overlapping sliding windows to segment the time series data, where we set the window size as 5120, as in [45]. As mentioned above, KAt dataset has 3 classes - 1 normal class (undamaged) and 2 faulty classes including inner faults and outer faults, which can be caused by either artificial or real damages. In this paper, we focused on the faults from real damages and generated 4900, 6200 and 6200 samples for normal class, inner faults and outer faults, respectively.

In addition, KAt bearing data was collected under 4 different working conditions, denoted as E, F, G and H. Table IV shows the parameter settings (i.e., rotational speed, load torque and radial force) for each working condition.

TABLE IV
DIFFERENT WORKING CONDITIONS

Working Condition	Rotational Speed [rpm]	Load Torque [Nm]	Radial Force [N]
E	900	0.7	1000
F	1500	0.1	1000
G	1500	0.7	400
H	1500	0.7	1000

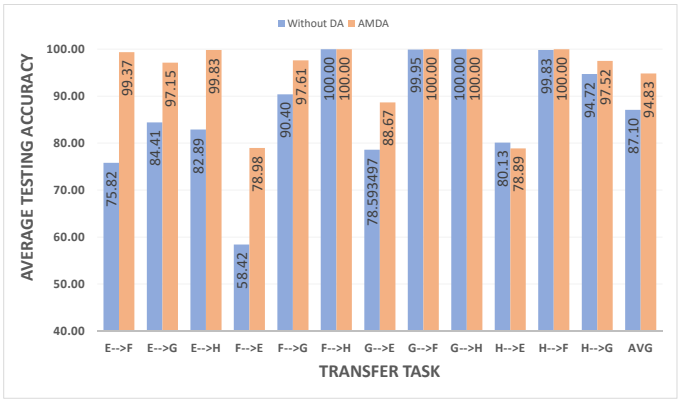

Fig. 6. Evaluation of AMDA with and without domain adaptation on KAt dataset using 12 cross-domain scenarios

2) Experimental Results: We also conducted 12 cross-domain experiments on KAt dataset to validate the performance of our proposed AMDA model. For example, we employed the working condition E as source domain and F, G and H as multiple target domains to generate the results for cross-domain tasks E→F, E→G and E→H.

Fig. 6 shows the evaluation results of our AMDA model with and without domain adaptation (without DA). Over 12

TABLE V
EVALUATION OF AMDA ON KAT DATASET AGAINST DOMAIN ADAPTATION BASELINES USING 12 CROSS-DOMAIN SCENARIOS

	Method	E→F	E→G	E→H	F→E	F→G	F→H	G→E	G→F	G→H	H→E	H→F	H→G	AVG
Shallow	CORAL	55.77	66.24	56.25	43.42	79.81	87.26	50.51	88.91	87.46	34.81	94.11	77.68	68.52
	TCA	42.06	68.47	45.36	53.00	80.56	93.42	63.36	92.26	90.62	51.59	96.96	84.45	71.84
	JDA	65.21	70.60	64.90	80.07	74.05	82.03	85.26	87.10	82.50	74.89	91.14	74.88	77.72
Deep	DDC	49.77	60.33	59.31	59.14	**97.84**	99.80	**89.14**	94.94	99.69	**86.07**	99.87	97.62	82.79
	Deep MMD	81.39	84.16	91.04	81.18	97.83	99.98	81.63	99.67	99.97	89.14	99.66	97.72	91.95
	Deep CORAL	84.06	87.03	88.80	**80.65**	90.17	99.99	83.22	99.98	99.99	80.44	100.00	**98.50**	91.07
	AMDA	**99.37**	**97.15**	**99.83**	78.98	97.61	**100.00**	88.67	**100.00**	**100.00**	78.89	**100.00**	97.52	**94.83**

TABLE VI
COMPARISON WITH STATE-OF-THE-ART METHODS

Method	F→G	F→H	G→F	G→H	H→F	H→G	AVG
ACDIN	79.43	78.73	85.07	90.53	79.53	75.60	81.48
WDCNN	72.33	94.70	69.33	69.77	93.67	70.27	78.35
Alexnet	78.87	98.47	65.93	66.20	96.03	74.07	79.93
Resnet	71.33	96.67	64.53	67.23	92.73	72.60	77.52
ICN	80.67	96.97	70.23	70.67	94.27	79.50	82.05
AMDA	**97.61**	**100.00**	**100.00**	**100.00**	**100.00**	**97.52**	**99.19**

cross-domain tasks, AMDA achieves an average accuracy of 94.83%, which is 7.73% higher than without DA. Once again, it demonstrates that the designed domain adaptation technique in AMDA model is effective for cross-domain fault diagnosis.

As shown in Fig. 6, without DA achieves relatively low performance for the 6 tasks involving the working condition E (i.e., E→F, E→G, E→H, F→E, G→E and H→E) with an average accuracy of 76.71%. This indicates that changing the rotational speed would cause more significant domain shift than changing load torque or radial force, leading to a large domain discrepancy between E and other 3 domains F, G and H. However, our AMDA can perform very well for these 6 *hard* transfer tasks - it achieves an average accuracy of 90.48%, with a significant improvement of 13.77% over without DA.

3) Comparison with Domain Adaptation Baselines: Here, we compare the proposed AMDA method with the same domain adaptation baselines, i.e., TCA, JDA, CORAL, DDC, Deep MMD, and Deep CORAL. Table V presents the comprehensive evaluation of various methods across 12 different transfer tasks. Over 8 out of 12 cross-domain tasks, our AMDA method performs better than the implemented baselines. Overall, AMDA achieves the highest average accuracy of 94.83% as shown in Table V, which is 3.76% higher than the second best method, i.e., Deep CORAL.

4) Comparison with the state-of-the-arts: The authors in [45] reported the performance of 5 deep learning based methods on Kat dataset using 6 cross-domain scenarios. These 5 state-of-the-art methods include ACDIN [46], WDCNN [34], AlexNet [47], ResNet [48] and ICN [45]. In particular, AlexNet and ResNet, which are famous convolutional architectures for image classification, were applied for fault diagnosis in [45]. Meanwhile, the other three methods are recently proposed for fault diagnosis. For example, ACDIN [46] refers to deep inception network with atrous convolution. The inception part in ACDIN concatenates multiple filters with different size to support different resolutions, while atrous convolution is a dilated filter to support wider input field. WDCNN [34] implements five 1-dimensional convolutional layers with wide input kernel. ICN [45] is an inception based capsule network for fault diagnosis, where the capsule network [49] is employed to capture correlation between different features and inception is used to extract features on different resolutions.

For fair comparison, we selected the same cross-domain scenarios for AMDA and the 5 state-of-the-arts above. Table VI shows the performance of various methods over 6 transfer tasks on KAt dataset. Overall, our AMDA significantly surpass the 5 competing approaches with an average accuracy of 97.52%, which is 15.47% higher than ICN (the second-best method).

D. Case 3: Self-collected Dataset

1) Dataset Description: We collected an additional dataset based on drivetrain dynamic simulator (DDS) platform [50] for further verification. The sampling rate of the vibration signal is 5120Hz. For this dataset, it consists of one normal class and three types of faults, i.e., inner-race (IR), outer-race (OR) and ball-crack (BC), under three different working conditions as shown in Table VII. We also use sliding windows with overlaps to segment the data, while the window size and the step size are the same as the CWRU dataset.

TABLE VII
DIFFERENT WORKING CONDITIONS FOR THE SELF-COLLECTED DATASET

Working Condition	Loading Torque	Fault Type
I	0 Nm	Normal, IR, OR, BC
J	7.2 Nm	Normal, IR, OR, BC
K	14.4 Nm	Normal, IR, OR, BC

2) Experimental Results: We denote three working conditions as I, J and K, which correspond to load 0 Nm, 7.2 Nm and 14.4 Nm respectively. Thus, six cross-domain experiments for our proposed method with and without DA

TABLE VIII
COMPARISON AGAINST DOMAIN ADAPTATION BASELINES

	Method	I→J	I→K	J→I	J→K	K→I	K→J	AVG
Shallow	CORAL	44.95	60.37	50.48	49.95	59.42	42.13	51.22
	TCA	74.30	49.61	87.52	50.19	56.37	58.67	62.78
	JDA	71.96	48.19	75.03	56.79	50.22	57.06	59.875
Deep	DDC	83.81	72.41	90.25	57.45	69.28	77.56	75.13
	Deep MMD	87.4	68.34	80.97	55.13	59.16	66.96	69.66
	Deep CORAL	89.45	68.01	87.49	61.91	65.20	68.84	73.48
	AMDA	**92.42**	**73.04**	**93.15**	**74.6**	**94.17**	**93.44**	**86.80**

have been performed as shown in Fig. 7. It is consistent with our previous evaluation that the DA can significantly improve the performance of fault diagnosis. Specifically, the proposed AMDA achieves an average accuracy of 86.80%, which is 11.50% higher than that without DA. This further indicates the effectiveness of the proposed method for cross-domain fault diagnosis.

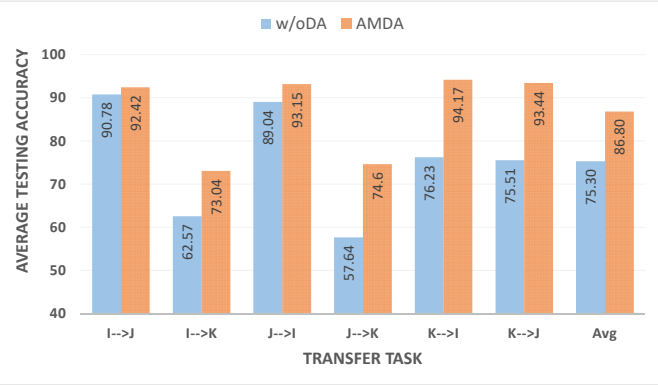

Fig. 7. Evaluation of AMDA with and without domain adaptation on self-collected dataset.

3) Comparison with Domain Adaptation Baselines: Similar to the previous evaluation, we compare with some advanced benchmark approaches for DA, including conventional DA methods (i.e., TCA, JDA and CORAL) and deep DA methods (i.e., DDC, Deep MMD and Deep CORAL). The results for the six cross-domain experiments are demonstrated in Table VIII. Due to the relatively large gap (load variation) between domains, the performances of all the approaches degrade to some extent. Consistently, our AMDA method outperforms the benchmark approaches in all the six cross-domain scenarios.

E. Evaluation of Proposed 1SmT Setting

In this section, we compare the 1SmT setting and 1S1T setting on KAt dataset in terms of generalization and time efficiency. For 1S1T, we selected the DDC method as it is the best baseline as shown in both Tables II and V. In addition to 1SmT and 1S1T settings, we further constructed 1SmxT setting by mixing N target domains as a single target domain. We also ran DDC and our AMDA under 1SmxT setting and included their results for comparison.

Table IX illustrates the accuracy of AMDA and DDC under different settings. The column E_source in Table IX means that E is used as source domain and F, G, and H are target domains (similarly for columns F_source, G_source and H_source). Clearly, our AMDA (1SmT) outperforms AMDA (1SmxT) by 3.38% and also significantly outperforms DDC under both 1S1T and 1SmxT settings. For DDC itself, mixing the target domains, i.e., DDC (1SmxT), leads to a performance deterioration of 6.84% compared to DDC (1S1T).

TABLE IX
ACCURACY (%) OF AMDA AND DDC UNDER DIFFERENT SETTINGS

	E_source	F_source	G_source	H_source	AVG
AMDA (1SmT)	98.78	92.20	96.22	92.14	**94.83**
AMDA (1S1T)	97.94	95.35	96.33	98.81	**97.11**
AMDA (1SmxT)	93.66	92.13	92.96	87.05	91.45
DDC (1SmxT)	45.78	80.94	92.70	84.37	75.95
DDC (1S1T)	56.47	85.59	94.59	94.52	82.79

TABLE X
TRAINING TIME (SEC) OF AMDA UNDER 1SMT AND 1S1T SETTINGS

Model	Total Time
AMDA (1SmT)	**712.07**
AMDA (1S1T)	**1781.12**

In addition, we can observe that AMDA (1S1T)–which is also our implementation–achieves higher accuracy than AMDA (1SmT). However, AMDA (1SmT) has higher scalability than AMDA (1S1T) and can generalize well to multiple target domains. In particular, AMDA (1SmT) can significantly reduce the model training compared with AMDA (1S1T), as shown in Table X. Therefore, our proposed AMDA (1SmT) is more suitable than AMDA (1S1T) for practical scenarios.

V. CONCLUSION

In this paper, we have introduced a novel domain adaptation scenario, i.e., single source multiple target (1SmT) setting, for fault diagnosis applications. It is more realistic than the existing single source single target (1S1T) setting, as working conditions may change in practice for manufacturing environments. We have proposed a novel adversarial multiple-target domain adaptation (AMDA) framework, which has a

deep learning architecture for adversarial unsupervised domain adaptation. Extensive experiments have been conducted to evaluate our proposed AMDA model on two public datasets and one self-collected dataset. Experimental results demonstrate that the proposed AMDA method significantly outperforms the benchmarking methods for cross-domain fault diagnosis. In our future works, we aim to extend domain adaption to include more physical variations. Moreover, the more challenging and practical domain adaption scenarios, such as cross environments or machines, will also be considered.

REFERENCES

[1] T. W. Rauber, F. de Assis Boldt, and F. M. Varejão, "Heterogeneous feature models and feature selection applied to bearing fault diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 637–646, 2015.

[2] Z. Chen, K. Gryllias, and W. Li, "Intelligent fault diagnosis for rotary machinery using transferable convolutional neural network," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2019.

[3] L. Wen, X. Li, and L. Gao, "A new two-level hierarchical diagnosis network based on convolutional neural network," *IEEE Transactions on Instrumentation and Measurement*, 2019.

[4] M. Sohaib and J.-M. Kim, "Fault diagnosis of rotary machine bearings under inconsistentworking conditions," *IEEE Transactions on Instrumentation and Measurement*, 2019.

[5] J. Sun, C. Yan, and J. Wen, "Intelligent bearing fault diagnosis method combining compressed data acquisition and deep learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 1, pp. 185–195, Jan 2018.

[6] H. Wang, J. Xu, R. Yan, and R. X. Gao, "A new intelligent bearing fault diagnosis method using sdp representation and se-cnn," *IEEE Transactions on Instrumentation and Measurement*, 2019.

[7] P. Liang, C. Deng, J. Wu, G. Li, Z. Yang, and Y. Wang, "Intelligent fault diagnosis via semi-supervised generative adversarial nets and wavelet transform," *IEEE Transactions on Instrumentation and Measurement*, 2019.

[8] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213–237, 2019.

[9] Q. Li, "Literature survey: domain adaptation algorithms for natural language processing," *Department of Computer Science The Graduate Center, The City University of New York*, pp. 8–10, 2012.

[10] X. Li, W. Zhang, and Q. Ding, "Cross-domain fault diagnosis of rolling element bearings using deep generative neural networks," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 7, pp. 5525–5534, 2018.

[11] Y. L. S. X. T. Y. L. Guo and N. Li, "Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data," *IEEE Transactions on Industrial Electronics.*, 2018.

[12] X. Li, W. Zhang, Q. Ding, and X. Li, "Diagnosing rotating machines with weakly supervised data using deep transfer learning," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1688–1697, 2020.

[13] Y. Song, Y. Li, L. Jia, and M. Qiu, "Retraining strategy based domain adaption network for intelligent fault diagnosis," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2019.

[14] C. Chen, F. Shen, J. Xu, and R. Yan, "Probabilistic latent semantic analysis based gear fault diagnosis under variable working conditions," *IEEE Transactions on Instrumentation and Measurement*, 2019.

[15] X. Li, W. Zhang, and Q. Ding, "A robust intelligent fault diagnosis method for rolling element bearings based on deep distance metric learning," *Neurocomputing*, vol. 310, pp. 77–95, 2018.

[16] W. Mao, J. He, and M. J. Zuo, "Predicting remaining useful life of rolling bearings based on deep feature representation and transfer learning," *IEEE Transactions on Instrumentation and Measurement*, 2019.

[17] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.

[18] R. Yan, F. Shen, C. Sun, and X. Chen, "Knowledge transfer for rotary machine fault diagnosis," *IEEE Sensors Journal*, 2019.

[19] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, "Deep model based domain adaptation for fault diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2296–2305, 2017.

[20] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Advances in neural information processing systems*, 2007, pp. 513–520.

[21] 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, 2017.

[22] L. Wen, L. Gao, and X. Li, "A new deep transfer learning based on sparse auto-encoder for fault diagnosis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[23] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 723–773, 2012.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[25] W. Qian, S. Li, and J. Wang, "A new transfer learning method and its application on rotating machine fault diagnosis under variant working conditions," *IEEE Access*, vol. 6, pp. 69 907–69 917, 2018.

[26] X. Li, W. Zhang, N.-X. Xu, and Q. Ding, "Deep learning-based machinery fault diagnostics with domain adaptation across sensors at different places," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2019.

[27] X. Wang, H. He, and L. Li, "A hierarchical deep domain adaptation approach for fault diagnosis of power plant thermal system," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 5139–5148, 2019.

[28] B. Zhang, W. Li, X.-L. Li, and S.-K. Ng, "Intelligent fault diagnosis under varying working conditions based on domain adaptive convolutional neural networks," *IEEE Access*, vol. 6, pp. 66 367–66 384, 2018.

[29] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8789–8797.

[30] A. Anoosheh, E. Agustsson, R. Timofte, and L. Van Gool, "Combogan: Unrestrained scalability for image domain translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 783–790.

[31] Y. Li, X. Tian, T. Liu, and D. Tao, "On better exploring and exploiting task relationships in multitask learning: Joint model and feature learning," *IEEE Transactions on Neural Networks*, vol. 29, no. 5, pp. 1975–1985, 2018.

[32] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[33] W. A. Smith and R. B. Randall, "Rolling element bearing diagnostics using the case western reserve university data: A benchmark study," *Mechanical Systems and Signal Processing*, vol. 64, pp. 100–131, 2015.

[34] 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, pp. 425–425, 2017.

[35] G.-Q. Jiang, P. Xie, X. Wang, M. Chen, and Q. He, "Intelligent fault diagnosis of rotary machinery based on unsupervised multiscale representation learning," *Chinese Journal of Mechanical Engineering*, vol. 30, no. 6, pp. 1314–1324, 2017.

[36] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2010.

[37] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2200–2207.

[38] B. Sun, J. Feng, and K. Saenko, "Correlation alignment for unsupervised domain adaptation," in *Domain Adaptation in Computer Vision Applications*. Springer, 2017, pp. 153–171.

[39] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[40] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2272–2281.

[41] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European conference on computer vision*. Springer, 2016, pp. 443–450.

[42] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of

rotating machinery with massive data," *Mechanical Systems and Signal Processing*, vol. 72, pp. 303–315, 2016.

[43] W. Zhang, C. Li, G. Peng, Y. Chen, and Z. Zhang, "A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load," *Mechanical Systems and Signal Processing*, vol. 100, pp. 439–453, 2018.

[44] C. Lessmeier, J. K. Kimotho, D. Zimmer, and W. Sextro, "Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: a benchmark data set for data-driven classification," in *Proceedings of the European conference of the prognostics and health management society*, 2016, pp. 05–08.

[45] Z. Zhu, G. Peng, Y. Chen, and H. Gao, "A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis," *Neurocomputing*, vol. 323, pp. 62–75, 2019.

[46] Y. Chen, G. Peng, C. Xie, W. Zhang, C. Li, and S. Liu, "Acdin: Bridging the gap between artificial and real bearing damages for bearing fault diagnosis," *Neurocomputing*, vol. 294, pp. 61–71, 2018.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[49] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in neural information processing systems*, 2017, pp. 3856–3866.

[50] F. Shen, R. Langari, and R. Yan, "Transfer between multiple machine plants: A modified fast self-organizing feature map and two-order selective ensemble based fault diagnosis strategy," *Measurement*, vol. 151, p. 107155, 2020.