Adversarial Multiple-Target Domain Adaptation for Fault Classification

Mohamed Ragab, *Graduate Student Member, IEEE*, Zhenghua Chen, *Member, IEEE*, Min Wu, *Member, IEEE*, Haoliang Li, *Member, IEEE*, Chee-Keong Kwoh, Ruqiang Yan, *Senior Member, IEEE*, and Xiaoli Li, *Senior Member, IEEE*

***Abstract*— Data-driven fault classification 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 deteriorates the diagnosis performance. Recently, domain adaptation (DA) 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 needs 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-target DA (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 domain 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 at https://github.com/mohamedr002/AMDA.**

***Index Terms*— Adversarial domain adaptation (DA), convolutional neural network (CNN), discriminator, intelligent fault diagnosis, single-source multiple-targets (1SmTs).**

I. Introduction

DATA-DRIVEN fault classification methods have the 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, and so on. While traditional machine learning techniques have been employed for machine fault diagnosis [1], they suffer from labor-intensive feature engineering and require a large amount of manually labeled training data.

During the 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, and natural language processing. Recently, deep learning has also been applied for fault classification. Chen *et al.* [2] employed 1-D convolutional neural network (CNN) with transferable features to leverage knowledge from the source domain for fault diagnosis of rotary machinery, while Wen *et al.* [3] developed a hierarchical diagnosis approach based on CNN to diagnose the fault and find its degradation level concurrently. Sohaib and Kim [4] integrated CNN with bispectrum analysis to achieve fault diagnosis of inconsistent working environments. 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.* [6] integrated CNN with squeeze and excitation networks to graphically represent the bearing states. Liang *et al.* [7] employed a semisupervised generative adversarial network coupled with wavelet transform to reduce the number of labeled samples.

Zhao *et al.* [8] performed a comprehensive review of different deep learning algorithms for fault diagnosis. Nevertheless, these methods work under the assumption that labeled 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 tests 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 of the unlabeled test data usually does not follow

Manuscript received January 24, 2020; accepted June 28, 2020. Date of publication July 15, 2020; date of current version November 23, 2020. This work was supported in part by the A*STAR Industrial Internet of Things Research Program though the RIE2020 IAF-PP under Grant A1788a0023 and in part by the National Natural Science Foundation of China under Grant 51835009. The work of Mohamed Ragab was supported by A*STAR SINGA Scholarship. The Associate Editor coordinating the review process was Lihui Peng. *(Corresponding author: Zhenghua Chen.)*

Mohamed Ragab and Chee-Keong Kwoh are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: mohamedr002@e.ntu.edu.sg; asckkwoh@ntu.edu.sg).

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

Haoliang Li is with the School of Electrical and Electronic and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: hli016@e.ntu.edu.sg).

Ruqiang Yan is with the School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: yanruqiang@xjtu.edu.cn).

Digital Object Identifier 10.1109/TIM.2020.3009341

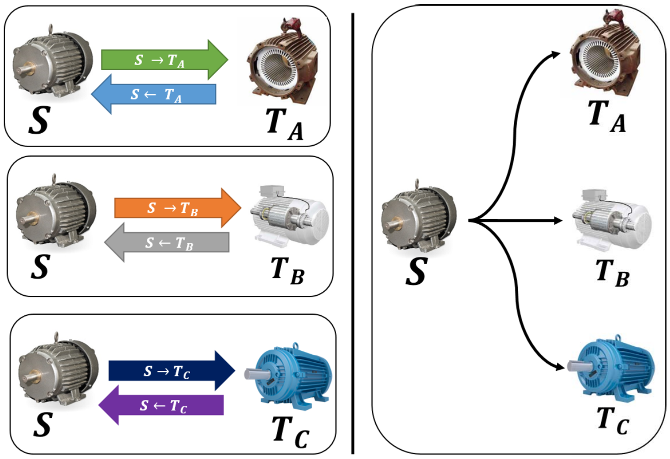

Fig. 1. Existing approaches versus our scalable multitarget approach.

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 customized 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 labeled 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 the domain-shift problem to transfer the model from the source domain (one working condition) to target domain (different working conditions) in intelligent fault diagnosis problems [10]–[12]. However, all existing methods work under single-source single-target (1S1T) settings, 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 article, we build upon the work done by Tzeng *et al.* [13] who proposed adversarial DA 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 DA problem under the single-source and multiple-targets (1SmTs) setting for fault diagnosis purposes. 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 DA 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-target 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.

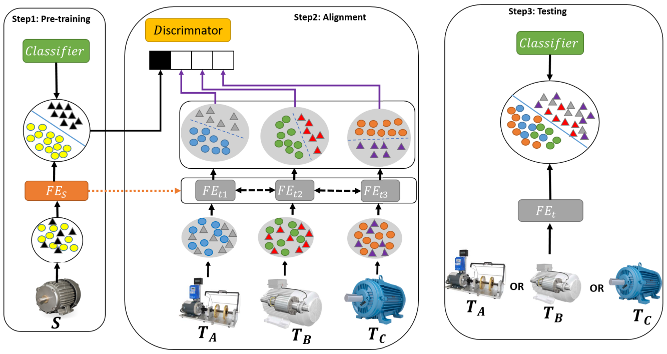

Fig. 2. Proposed AMDA for fault diagnosis.

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

1) We formulate a more realistic 1SmT problem that is particularly used for real-world fault diagnostic problems.
2) We propose a novel adversarial multiple-target DA (AMDA) method that designs a deep learning architecture for adversarial unsupervised DA to address the 1SmT problem. To the best of our knowledge, it is the first attempt in this area.
3) We addressed the limited scalability of existing approaches by proposing a general model that can generalize to multiple targets concurrently.
4) 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 DA transfers knowledge to the 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 the 1S1T scenario, which can only handle a single-target domain at a time. Differently, we propose a novel 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 DA for fault diagnosis using 1S1T scenario iteyan2019knowledge. In [15],

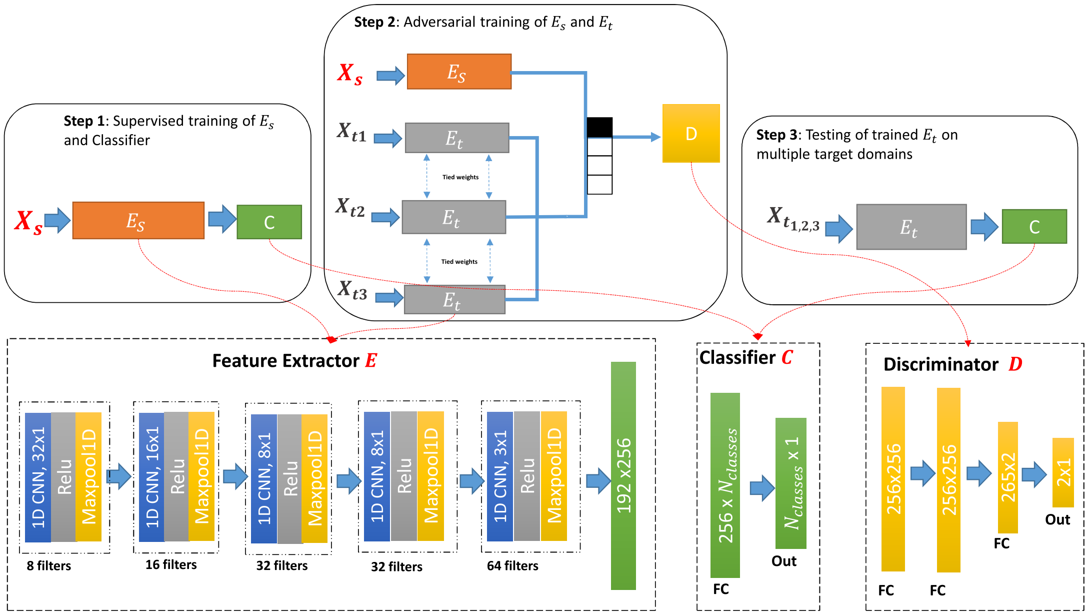

Fig. 3. Adversarial 1SmT DA (AMDA) model for fault classification with three main architectures: feature extractors (e.g., E_s for source domain and E_t for target domains), classifier C, and discriminator D.

researchers employed autoencoder to extract domain-invariant features, with the help of popular domain discrepancy metric maximum mean discrepancy (MMD) [16] to measure the discrepancy between the source and target distributions. Minimizing both autoencoder 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.* [17]. Very recently, Li *et al.* [18] employed 1-D CNN to extract feature representation from frequency-domain features. They also used a representation clustering scheme to maximize intraclass similarity and reduce interclass similarity, coupled with classification loss for more discriminative features and adopted MMD to obtain domain-invariant features [19]. Song *et al.* [12] proposed a DA network (DAN) with a retraining strategy based on pseudolabels to minimize the discrepancy between the source and target domains. Li *et al.* [20] proposed a DA approach to address fault diagnosis problems with data from different places in the same machine. Particularly, they integrated a gradient reversal layer with a novel parallel data alignment technique to tackle the domain-shift problem. In [21], a hierarchical deep DA approach has been used for fault diagnosis of the thermal system under varying working conditions. Especially, they employed correlation alignment (CORAL) with successive denoising autoencoders to learn domain-invariant features among different working conditions.

In [22], a two-phase approach was proposed, where the authors first pretrained a model on the source-domain data using 1-D CNN and then fine-tuned the untied model using target-domain data and MMD. Shao *et al.* [23] leveraged a pretrained network for the extraction of low-level features while using wavelet transformation with time-frequency representation of the data to fine-tune the model. In [24], an online fault diagnosis approach has been developed based on a transferable CNN and image representation of time-domain signals. Xing *et al.* [25] developed a deep belief network with MMD to obtain distribution invariant features. Li *et al.* [26] proposed a multikernel MMD across multiple layers to align the source and target distributions.

Yet, these approaches have only considered a single-target domain at a time. Hence, the model will have limited scalability by only generalizing to a single-target domain at a time, and one needs to train a new model independently for each target domain. Different from existing approaches (see Fig. 3), we tie the weights of the feature extractors of the multiple-target domains, inspired by multitask learning [27]. This enables a single feature extractor to generalize to multiple-target domains during the testing stage. In addition, it helps to reduce the capacity of the model and acts as a regularize to avoid overfitting. To this end, unlike all existing approaches, which can generalize to a single target at a time, our model can be more scalable and has a generalization ability that can handle multiple targets concurrently.

B. Single-Source Multiple-Targets

Among the DA literature, a little attention has been paid to (1SmT) problem. Recently, some approaches have addressed multiple domain learning problems [28]. However, they all in the context of image generation tasks, where they train a single generator to generate samples from different domains. Differently, our AMDA approach is addressing (1SmT) for the time-series classification problem. To the best of our knowledge, our proposed AMDA is the first trial in this application.

III. Adversarial Multiple-Target-Domain Adaptation

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

Different from the existing approaches, we tie the weights of the feature extractors of the multiple-target domains, inspired by multitask learning [27]. This enables a single feature extractor to generalize to multiple-target domains during the 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 a 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; and 3) test the domain adapted model on all N target domains. The goal of this article 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 and target domains is minimized. As such, the model can be better generalized to the multiple-target domains concurrently. In the following, we will explain each step in more detail.

A. Problem Formulation

The DA involves a domain $\mathcal{D}$ and task $\mathcal{T}$ [29], 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.

1) We have a labeled 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 ith 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.
2) The feature space of the 1SmT domains is same, i.e., $\mathcal{X}_s = \mathcal{X}_{t(1)} = \mathcal{X}_{t(2)} = \cdots = \mathcal{X}_{t(N)}$, where N is the number of target domains.
3) The marginal distribution between the source and target domains is different due to variation on multiple-target domains (e.g., with different working conditions), i.e., $P_s(\mathbf{x}) \neq P_{t(j)}(\mathbf{x})$ $(j = 1, 2, \ldots, N)$. 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 $j \neq k$.
4) Label space of the single-source domain and multiple-target domains is the same, i.e., $\mathcal{Y}_s = \mathcal{Y}_{t(1)} = \mathcal{Y}_{t(2)} = \cdots = \mathcal{Y}_{t(N)}$.

B. Supervised Learning With Labeled Source-Domain Data

Our first step employs the labeled source-domain data $\mathcal{D}_s = \{\mathbf{x}_s^i, y_s^i\}_{i=1}^{n_s}$, where $y_s^i \in \{1, \ldots, k\}$ 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:

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

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 the 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 pseudocode, 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 (1).

Algorithm 1 Supervised Learning Using Labeled 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. $X_s \leftarrow \{\mathbf{x}_s^1, \ldots, \mathbf{x}_s^m\}$, mini-batch of source samples
    2. $\mathbf{y}_s \leftarrow \{y_s^1, \ldots, y_s^m\}$, mini-batch of source labels
    3. $Preds \leftarrow C(E_s(X_s))$
    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 are invariant from the source-domain features [30]. 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 a 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 the training phase. As a result, we can use the common weights of target feature extractors to map any of the 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 $\{X_{t(1)}, \ldots, X_{t(N)}\}$ 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-domain 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 (GANs) loss [30]. 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 decides whether it belongs to the source domain or the target domains, under standard supervised learning fashion, where the loss is denoted as $\mathcal{L}_D$

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

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

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

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

where $E_{t(i)}$ is the feature extractor for the ith 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 $\mathcal{L}_E$.

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: $\{\mathcal{D}_{t(1)}, \ldots, \mathcal{D}_{t(N)}\}$, 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
        $E_{t(1)}, \ldots, E_{t(N)}$
$E_s \leftarrow$ 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 $X_s \sim P_s$
    2. Sample mini-batch of $m$ from each target domain:
       $\{X_{t(1)}, \ldots, X_{t(N)}\} \sim \{P_{t(1)}, \ldots P_{t(N)}\}$
    3. Extract source-domain features: $E_s(X_s)$
    4. Extract features from N target domains concurrently:
       $\{E_{t(1)}(X_{t(1)}), \ldots, E_{t(N)}(X_{t(N)})\}$
    5. Update $D$ by Eq. 2 // Train Discriminator
    for M steps do // Train $E_t$ M times
        6. Extract features from N target domains:
           $\{E_{t(1)}(X_{t(1)}), \ldots, E_{t(N)}(X_{t(N)})\}$
        7. Update the target feature extractor $E_t$ by Eq. 3
    end
end
```

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. Equation (4) shows the usage of softmax to compute the probability of each class given the input instance from any target domains

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

where $\mathbf{f}_t$ is latent representation of the corresponding target domain, and $C_{k'}(\cdot)$ denotes the output of kth class resulted from softmax.

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 1-D CNN as a feature extractor and used a wide input kernel for longer dependencies. A fully connected neural network with a softmax layer was used for fault classification, while a two-layer fully connected network was used to discriminate between

TABLE I
CWRU Bearing Dataset Description [32]

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

the source-and target-domain data. Fig. 3 shows 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) [31] benchmark dataset, which has been collected from the drive end of the 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 3 hp. Each subset has four 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 the centered position of @6:00 relative to the load zone. Moreover, each type of fault could have three different fault sizes, i.e., 0.007, 0.014, and 0.021 in, which leads to ten different classes (one normal class and nine 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 [17]. The corresponding window width and shifting step are 4096 and 295, respectively. Eventually, each working condition has 4000 samples, and each sample is represented as a 4096-D vector.

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 three experiments (A→B, A→C, and A→D), we used working condition A as the 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 domains 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 DA in our model for cross-domain fault classification.

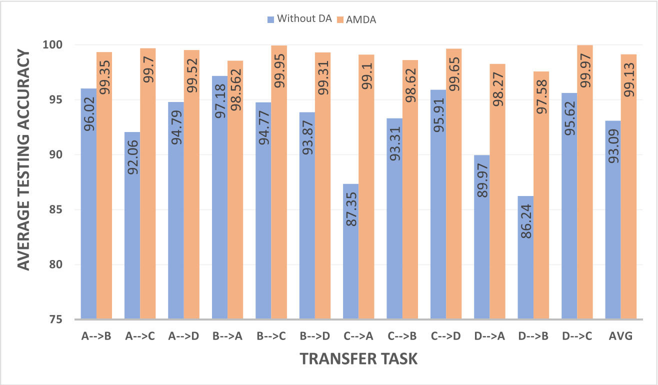

Fig. 4. Evaluation of AMDA with and without DA on CWRU dataset using 12 cross-domain scenarios.

Note that we use a one-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 DA Baselines: To demonstrate the superiority of the proposed AMDA, we implemented four DA baselines: transfer component analysis (TCA) [33], joint distribution adaptation (JDA) [34], CORAL [35], deep domain confusion (DDC) [36], deep MMD [37], and Deep CORAL [38].

Table II shows the results of different DA techniques using the 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 DA scenarios with an overall accuracy of 99.13%, which indicates the effectiveness of the proposed AMDA for this DA 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 three different state-of-the-art baselines, which are summarized as follows.

1) The first approach is fault diagnosis using deep neural network (DNN) [39], which consists of pretraining the stacked-autoencoder in an unsupervised manner and fine-tuning the network under the supervision of source labels.
2) The second approach is a five-layer CNN with a wide input kernel that was demonstrated to achieve high accuracy (WDCNN) [17].
3) The third approach is transfer inference with CNN (TICNN) with a six-layer CNN and introduces dropout

TABLE II

EVALUATION OF AMDA ON CWRU DATASET AGAINST DA 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 SIX 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
FDGN	97.81	96.81	97.27	96.32	95.44	96.55	96.70
AMDA	**99.4**	**99.7**	**98.6**	**99.9**	**99.1**	**98.6**	**99.21**

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

in the first input layer. In addition, ensemble learning has been employed to stabilize the performance of their model [40].

4) The last approach is fault diagnosis with generative networks (FDGN) [10], which employed GANs [30] to generate faulty data in the target domain and applied the generated data into the DA scheme to solve the cross-domain problem.

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 six cross-domain experiments. Therefore, we also conducted the same cross-domain experiments for a 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 DA, 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 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 [41]. 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 [41]. In this dataset, 32 experiments for rolling bearing elements were conducted to collect three types of data, namely, undamaged bearing data, artificially damaged bearing data, and real damaged bearing data. In particular, the bearing data in each experiment have 20 files, and each file was collected for 4 s with a sampling rate of 64 kHz.

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 [42]. As mentioned earlier, the KAt dataset has three classes—one normal class (undamaged) and two faulty classes, including inner faults and outer faults, which can be caused by either artificial or real damages. In this article, 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 were collected under four 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.

2) Experimental Results: We also conducted 12 cross-domain experiments on the KAt dataset to validate the performance of our proposed AMDA model. For example, we employed the working condition E as the 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. 5 shows the evaluation results of our AMDA model with and without DA. Over 12 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

TABLE V

EVALUATION OF AMDA ON KAT DATASET AGAINST DA 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**

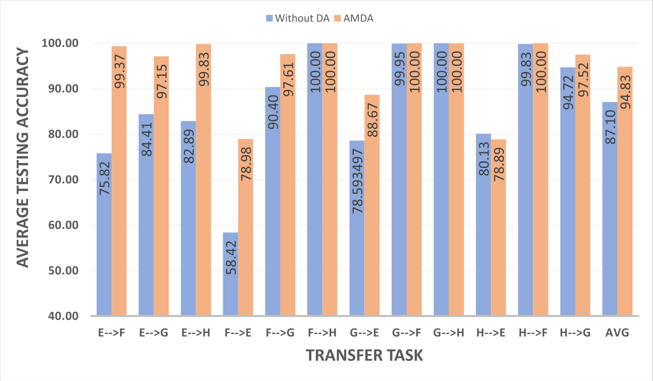

Fig. 5. Evaluation of AMDA with and without DA on the KAt dataset using 12 cross-domain scenarios.

TABLE VI

COMPARISON WITH STATE-OF-THE-ART METHODS

Method	**F→G**	**F→H**	**G→F**	**G→H**	**H→F**	**H→G**	**AVG**
DAN	85.70	98.40	81.58	89.29	98.00	90.50	90.58
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**

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

designed DA technique in the AMDA model is effective for cross-domain fault classification.

As shown in Fig. 5, without DA achieves relatively low performance for the six 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 three domains F, G, and H. However, our AMDA can perform very well for these six hard transfer tasks—it achieves an average accuracy of 90.48%, with a significant improvement of 13.77% over without DA.

3) Comparison With DA Baselines: Here, we compare the proposed AMDA method with the same DA baselines, i.e., TCA, JDA, CORAL, DDC, Deep MMD, and Deep CORAL. Table V presents a comprehensive evaluation of various methods across 12 different transfer tasks. Over eight 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: Zhu *et al.* [42] reported the performance of five deep learning based methods on KAt dataset using six cross-domain scenarios. These six state-of-the-art methods include DAN [12], ACDIN [43], WDCNN [17], AlexNet [44], ResNet [45], and ICN [42]. In particular, AlexNet and ResNet, which are famous convolutional architectures for image classification, were applied for fault diagnosis in [42]. Meanwhile, the other three methods are recently proposed for fault diagnosis. For example, ACDIN [43] 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 [17] implements five 1-D convolutional layers with wide input kernel. ICN [42] is an inception based capsule network for fault diagnosis, where the capsule network [46] 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 five state-of-the-arts above. Table VI shows the performance of various methods over six transfer tasks on KAt dataset. Overall, our AMDA significantly surpass the five competing approaches with an average accuracy of 97.52%, which is 15.47% higher than ICN (the second-best method).

TABLE VIII
COMPARISON AGAINST DA 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	59.88
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**

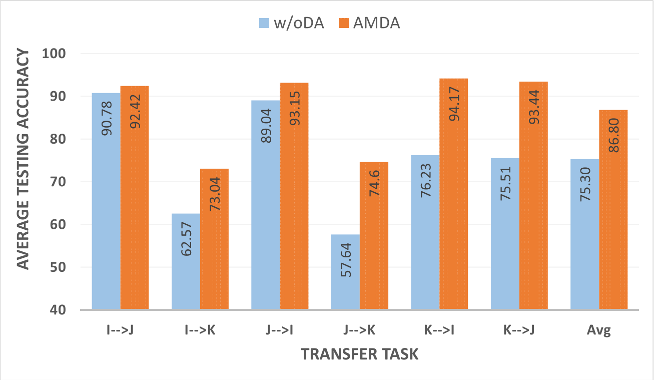

Fig. 6. Evaluation of AMDA with and without DA on the self-collected dataset.

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**

D. Case 3: Self-Collected Dataset

1) Dataset Description: We collected an additional dataset based on the drivetrain dynamic simulator (DDS) platform [47] for further verification. The sampling rate of the vibration signal is 5120 Hz. 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.

2) Experimental Results: We denote three working conditions as I, J, and K, which correspond to load 0, 7.2, and 14.4 Nm, respectively. Thus, six cross-domain experiments for our proposed method with and without DA have been performed, as shown in Fig. 6. It is consistent with our previous evaluation that the DA can significantly improve the performance of fault classification. Especially, 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 classification.

3) Comparison With DA 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 the 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 Tables II and V. In addition to 1SmT and 1S1T settings, we further constructed a 1SmxT setting by mixing N target domains as a single-target domain. We also ran DDC and our AMDA under the 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 the 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 performance deterioration of 6.84% compared with DDC (1S1T).

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 article, we have introduced a novel DA scenario, i.e., 1SmT setting, for fault classification applications. It is more realistic than the existing 1S1T setting, as working conditions may change in practice for manufacturing environments. We have proposed a novel AMDA framework, which has a deep learning architecture for adversarial unsupervised DA. 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 classification. 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 Trans. Ind. Electron.*, vol. 62, no. 1, pp. 637–646, Jan. 2015.

[2] Z. Chen, K. Gryllias, and W. Li, “Intelligent fault diagnosis for rotary machinery using transferable convolutional neural network,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 339–349, Jan. 2020.

[3] L. Wen, X. Li, and L. Gao, “A new two-level hierarchical diagnosis network based on convolutional neural network,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 2, pp. 330–338, Feb. 2020.

[4] M. Sohaib and J.-M. Kim, “Fault diagnosis of rotary machine bearings under inconsistent working conditions,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 6, pp. 3334–3347, Jun. 2020.

[5] J. Sun, C. Yan, and J. Wen, “Intelligent bearing fault diagnosis method combining compressed data acquisition and deep learning,” *IEEE Trans. Instrum. Meas.*, 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 Trans. Instrum. Meas.*, vol. 69, no. 5, pp. 2377–2389, May 2020.

[7] P. Liang, C. Deng, J. Wu, G. Li, Z. Yang, and Y. Wang, “Intelligent fault diagnosis via semisupervised generative adversarial nets and wavelet transform,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 4659–4671, Jul. 2020.

[8] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, “Deep learning and its applications to machine health monitoring,” *Mech. Syst. Signal Process.*, vol. 115, pp. 213–237, Jan. 2019.

[9] Q. Li, “Literature survey: Domain adaptation algorithms for natural language processing,” Ph.D. dissertation, Dept. Comput. Sci. The Graduate Center, City Univ. New York, New York, NY, USA, 2012, pp. 8–10.

[10] X. Li, W. Zhang, and Q. Ding, “Cross-domain fault diagnosis of rolling element bearings using deep generative neural networks,” *IEEE Trans. Ind. Electron.*, vol. 66, no. 7, pp. 5525–5534, Jul. 2019.

[11] L. Guo, Y. Lei, S. Xing, T. Yan, 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.

[12] Y. Song, Y. Li, L. Jia, and M. Qiu, “Retraining strategy-based domain adaption network for intelligent fault diagnosis,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6163–6171, Sep. 2020.

[13] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7167–7176.

[14] R. Yan, F. Shen, C. Sun, and X. Chen, “Knowledge transfer for rotary machine fault diagnosis,” *IEEE Sensors J.*, vol. 20, no. 15, pp. 8374–8393, Aug. 2020.

[15] 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.

[16] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, “A kernel method for the two-sample-problem,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 513–520.

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

[18] 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, Oct. 2018.

[19] 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.

[20] 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 Trans. Ind. Electron.*, vol. 67, no. 8, pp. 6785–6794, Aug. 2020.

[21] X. Wang, H. He, and L. Li, “A hierarchical deep domain adaptation approach for fault diagnosis of power plant thermal system,” *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 5139–5148, Sep. 2019.

[22] 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. 66367–66384, 2018.

[23] S. Shao, S. McAleer, R. Yan, and P. Baldi, “Highly accurate machine fault diagnosis using deep transfer learning,” *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2446–2455, Apr. 2019.

[24] G. Xu, M. Liu, Z. Jiang, W. Shen, and C. Huang, “Online fault diagnosis method based on transfer convolutional neural networks,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 2, pp. 509–520, Feb. 2020.

[25] S. Xing, Y. Lei, S. Wang, and F. Jia, “Distribution-invariant deep belief network for intelligent fault diagnosis of machines under new working conditions,” *IEEE Trans. Ind. Electron.*, early access, Feb. 13, 2020, doi: 10.1109/TIE.2020.2972461.

[26] 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.

[27] 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 Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1975–1985, May 2018.

[28] A. Anoosheh, E. Agustsson, R. Timofte, and L. Van Gool, “ComboGAN: Unrestrained scalability for image domain translation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2018, pp. 783–790.

[29] 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.

[30] I. Goodfellow *et al.*, “Generative adversarial nets,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[31] 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.*, vols. 64–65, pp. 100–131, Dec. 2015.

[32] G.-Q. Jiang, P. Xie, X. Wang, M. Chen, and Q. He, “Intelligent fault diagnosis of rotary machinery based on unsupervised multiscale representation learning,” *Chin. J. Mech. Eng.*, vol. 30, no. 6, pp. 1314–1324, Nov. 2017.

[33] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[34] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, “Transfer feature learning with joint distribution adaptation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2200–2207.

[35] B. Sun, J. Feng, and K. Saenko, “Correlation alignment for unsupervised domain adaptation,” in *Domain Adaptation in Computer Vision Applications*. Cham, Switzerland: Springer, 2017, pp. 153–171, doi: 10.1007/978-3-319-58347-1_8.

[36] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, “Deep domain confusion: Maximizing for domain invariance,” 2014, *arXiv:1412.3474*. [Online]. Available: http://arxiv.org/abs/1412.3474

[37] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2272–2281.

[38] B. Sun and K. Saenko, “Deep coral: Correlation alignment for deep domain adaptation,” in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 443–450.
[39] 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,” *Mech. Syst. Signal Process.*, vols. 72–73, pp. 303–315, May 2016.
[40] 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,” *Mech. Syst. Signal Process.*, vol. 100, pp. 439–453, Feb. 2018.
[41] 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 *Proc. Eur. Conf. Prognostics Health Manage. Soc.*, 2016, pp. 05–08.
[42] 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, Jan. 2019.
[43] 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, Jun. 2018.
[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[45] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[46] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic routing between capsules,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3856–3866.
[47] 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, Feb. 2020, Art. no. 107155.

Mohamed Ragab (Graduate Student Member, IEEE) received the B.Sc. degree (Hons.) and the M.Sc. degree from the Department of Electrical Engineering, Aswan University, Aswan, Egypt, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore.

He is with the Machine Intellection (MI) Department, Institute of Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore. His research interests include deep learning, transfer learning, and intelligent fault diagnosis and prognosis.

Zhenghua Chen (Member, IEEE) received the B.Eng. degree in mechatronics engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2011, and the Ph.D. degree in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2017.

He has been working with NTU as a Research Fellow. He is currently a Scientist with the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. His research interests include sensory data analytics, machine learning, deep learning, transfer learning, and related applications.

Min Wu (Member, IEEE) received the B.S. degree in computer science from the University of Science and Technology of China (USTC), Chengdu, China, in 2006, and the Ph.D. degree in computer science from Nanyang Technological University (NTU), Singapore, in 2011.

He is currently a Senior Scientist with the Data Analytics Department, Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. His current research interests include machine learning, data mining, and bioinformatics.

Dr. Wu received the best paper awards in InCoB 2016 and DASFAA 2015. He also won the IJCAI Competition on repeated buyers prediction in 2015.

Haoliang Li (Member, IEEE) received the B.Eng. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2013, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2018.

He was a Project Officer in 2018 and a Research Fellow from July 2018 to May 2019 at the Rapid-Rich Object Search Laboratory, NTU. He is currently a Wallenberg-NTU Presidential Post-Doctoral Fellow with NTU. His research interest is information forensics and security.

Dr. Li received the Best Thesis Doctorate Innovation Award from NTU and the Wallenberg-NTU Presidential Fellowship from Wallenberg Foundation, Sweden, and NTU in 2019.

Chee-Keong Kwoh received the bachelor’s degree (Hons.) in electrical engineering and the master’s degree in industrial system engineering from the National University of Singapore, Singapore, in 1987 and 1991, respectively, and the Ph.D. degree from the Imperial College of Science, Technology, and Medicine, University of London, London, U.K., in 1995.

He has been with the School of Computer Engineering, Nanyang Technological University (NTU), Singapore, since 1993. He is currently the Deputy Executive Director of PaCE, NTU. His research interests include data mining, soft computing, and graph-based inference and application areas include bioinformatics and engineering. He has done significant research work in his research areas and has published many quality international conferences and journal articles.

Dr. Kwoh is a member of the Association for Medical and Bio-Informatics and the Imperial College Alumni Association of Singapore. He has provided many services to professional bodies in Singapore and was conferred the Public Service Medal by the President of Singapore in 2008.

Ruqiang Yan (Senior Member, IEEE) received the M.S. degree in precision instrument and machinery from the University of Science and Technology of China, Hefei, China, in 2002, and the Ph.D. degree in mechanical engineering from the University of Massachusetts at Amherst, Amherst, MA, USA, in 2007.

From 2009 to 2018, he was a Professor with the School of Instrument Science and Engineering, Southeast University, Nanjing, China. He joined the School of Mechanical Engineering, Xi’an Jiaotong University, Xi’an, China, in 2018. He holds 28 patents. He has published two books and over 200 articles in technical journals and conference proceedings. His research interests include data analytics, machine learning, and energy-efficient sensing and sensor networks for the condition monitoring and health diagnosis of large-scale, complex, dynamical systems.

Dr. Yan is a fellow of ASME in 2019. His honors and awards include the IEEE Instrumentation and Measurement Society Technical Award in 2019, the New Century Excellent Talents in University Award from the Ministry of Education in China in 2009, and multiple best paper awards. He is also the Associate Editor-in-Chief of the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT and an Associate Editor of the IEEE SYSTEMS JOURNAL.

Xiaoli Li (Senior Member, IEEE) is currently a Principal Scientist with the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. He also holds adjunct professor positions at Nanyang Technological University, Singapore. He has published more than 200 high-quality articles. His research interests include data mining, machine learning, AI, and bioinformatics.

Prof. Li has been serving as a (senior) PC member/workshop chair/session chair in leading data mining and AI-related conferences, including KDD, ICDM, SDM, PKDD/ECML, WWW, IJCAI, AAAI, ACL, and CIKM. He has received numerous best paper/benchmark competition awards.