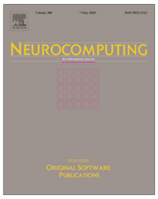
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Wasserstein distance based deep adversarial transfer learning for intelligent fault diagnosis with unlabeled or insufficient labeled data

Cheng Cheng [a], Beitong Zhou [a], Guijun Ma [b,c], Dongrui Wu [a], Ye Yuan [a,b,*]

[a] School of Artificial Intelligence and Automation, MOE Key Lab of Intelligent Control and Image Processing, Huazhong University of Science and Technology, Wuhan 430074, China
[b] State Key Lab of Digital Manufacturing Equipment and Technology, Wuhan 430074, China
[c] School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

ARTICLE INFO

Article history:
Received 27 January 2020
Revised 28 April 2020
Accepted 18 May 2020
Available online 26 May 2020
Communicated by Ma Lifeng Ma

Keywords:
Transfer learning
Domain adaptation
Wasserstein distance
Convolutional neural networks
Intelligent fault diagnosis

ABSTRACT

Intelligent fault diagnosis is one critical topic of maintenance solution for mechanical systems. Deep learning models, such as convolutional neural networks (CNNs), have been successfully applied to fault diagnosis tasks and achieved promising results. However, one is that two datasets (in source and target domains) of similar tasks are with different feature distributions because of different operational conditions; another one is that insufficient or unlabeled data in real industry applications (target domains) limit the adaptability of the source domain well-defined models. To solve the above problems, the concept of transfer learning should be adopted for domain adaptation, in the meantime, a network performs both supervised and unsupervised learning is required. Inspired by Wasserstein distance of optimal transport, in this paper, we propose a novel Wasserstein Distance-based Deep Transfer Learning (WD-DTL) network for both supervised and unsupervised fault diagnosis tasks. WD-DTL learns domain feature representations (generated by a CNN based feature extractor) and minimizes distributions between the source and target domains through an adversarial training process. The effectiveness of the proposed WD-DTL is verified through 16 different transfer tasks. Results show that WD-DTL achieves the highest diagnostic accuracies when compared to the existing Maximum Mean Discrepancy and CNN networks in almost all transfer tasks.

1. Introduction

Fault diagnosis aims to isolate faults on defective systems by monitoring and analyzing machine status using acquired measurements and other information, which requires experienced experts with a high skill set. This drives the demand for artificial intelligence techniques to make fault diagnosis decisions [1]. The deployment of a real-time fault diagnosis framework allows the maintenance team to act in advance to replace or fix the affected components, thus, to improve production efficiency and guarantee operational safety.

Over the past decade, many advanced signal processing and machine learning techniques have been used for fault diagnosis [2]. Signal processing techniques such as wavelet [3] and Hilbert-Huang transform [4] are adopt for feature extraction from faulty vibration signals, and machine learning models are then applied to automate the fault diagnosis procedure. In last few years, deep learning models, such as deep belief networks [5], sparse auto-encoder [6], and especially convolutional neural networks (CNN) [7], have shown superior fitting and learning abilities in fault diagnosis tasks over ruled-based and model-based methods. However, the above stated deep learning approaches suffer two difficulties: 1) Most of the approaches work well under the same hypothesis: the datasets for source domain and target domain tasks are required to be identically distributed. Thus, the adaptability of the pre-trained network is limited when facing new diagnosis tasks, where the different operational conditions and physical characteristics of the new task might cause distribution difference between the new dataset (target dataset) and the original dataset (source dataset). As a result, for a new fault diagnosis task, the deep learning model is commonly reconstructed from scratch, which results in the waste of computational resources and training time; 2) Insufficient labeled or unlabeled data in target domain is another common problem. In real industry situations, for a new diagnosis task, it is extremely difficult to collect sufficient typical samples to re-build a large-scale and high-quality dataset to train a network.

[*] Corresponding author at: School of Artificial Intelligence and Automation, MOE Key Lab of Intelligent Control and Image Processing, Huazhong University of Science and Technology, Wuhan 430074, China.
E-mail address: yye@hust.edu.cn (Y. Yuan).

https://doi.org/10.1016/j.neucom.2020.05.040
0925-2312/© 2020 Elsevier B.V. All rights reserved.

Deep transfer learning (DTL) [8] aims to perform learning in a target domain (with insufficient labeled or unlabeled data) by leveraging knowledge from relevant source domains (with sufficient labeled data), saving much expenditure on reconstructing a new fault diagnosis model from scratch and recollecting sufficient diagnosis labeled samples. Many successful approaches to DTL has been seen in various fields, including pattern recognition [9], image classification [10], and speech recognition [11].

Solutions to DTL can be roughly classified into three categories: instance-based DTL, network-based DTL, and mapping-based DTL. Instance-based DTL reweighs/subsamples a group of instances from the source domain to match the distributions in the target domain. Network-based DTL crops out part of the network pre-trained in the source domain, which is transferred to be a part of target networks for a relevant new task; see [12,13] for recent examples of instance-based and network-based DTL, respectively. However, the above approaches are not capable of learning a latent representation from the deep architecture. Mapping-based DTL, compared with other approaches to adapting deep models, has shown excellent properties through finding a common latent space, where the feature representations for source and target domains are invariant. Tzeng et al. [14] proposed a CNN architecture based network for domain adaptation, which introducing an adaptation layer to learn the feature representations. Maximum mean discrepancy (MMD) metric is used as an additional loss for the overall structure to compute the distribution distance with respect to a particular representation, which helps to select the depth and width of the architecture as well as to regulate the loss function during fine-tuning. Later, in [15,16], MMD was extended to multiple kernel variance MMD (MK-MMD) and joint MMD (JMMD) for better domain adaptation performance. However, the limitation of MMD method for domain adaptation is that the computational cost of MMD is quadratically increased with a large number of samples when calculating the Integral Probability Metrics (IPMs) [17]. Recently, Ajovsky et al. [18] indicate that Wasserstein distance can be a new direction to find better distribution mappings. Compared with other popular probability distances and divergences, such as Kullback–Leibler (KL) divergence and Jensen-Shannon (JS) divergence, [18] demonstrated that Wasserstein distance is a more sensible cost function when learning distributions supported by low dimensional manifolds. Later on, [19,20] proposed a new gradient penalty term for domain alignment critic parameters to solve the gradient vanishing or exploding problems in [18]. [21] extended [20] by using second-order terms in the form of the graph between the domain representations. Hence, the essence of our proposed approach is to adopt the Wasserstein distance to train a DTL model for intelligent fault diagnosis problem which seeks to minimize the distributions between the source domain and target domains. Our motivation for this work is to figure out how Wasserstein distance behaves in transfer learning due to its excellent performance in the generative adversarial network (GAN).

This paper concerns the problem of DTL modeling to explore the transferable features of fault diagnosis under different operating conditions. In the source domain, a base CNN model is first trained with sufficient data. Then, we build a Wasserstein distance-based DTL (WD-DTL) to learn invariant features between the source and target domains. A neural network is introduced (denoted by domain alignment critic) to calculate the empirical Wasserstein distance by maximizing domain alignment critic loss. After this procedure, a classifier is introduced to optimize the CNN-based feature extractor parameters by minimizing the estimated empirical Wasserstein distance. Through the above adversarial learning process, the transferable features from a source domain where faulty labels are known can be brought to diagnose a new but relevant diagnosis task without any labeled sample. Experimental results, through 16 transfer tasks, demonstrate the effectiveness of the distance measurement method and the proposed DTL model. This paper makes the following contributions: 1) Wasserstein distance is used as the distance measurement of domains in fault diagnosis problems to explore better distribution mapping, 2) the proposed WD-DTL framework could perform both unsupervised and supervised transfer tasks. Consequently, for a new diagnosis task, this is a novel approach which could contribute to solving both unlabeled and insufficient labeled data in real industrial applications, 3) the versatility of our WD-DTL approach is demonstrated with transfer learning experiments, in terms of 3 different transfer scenarios and 16 transfer tasks in total, and 4) the proposed WD-DTL approach surpasses the existing MMD, CNN, and other existing methods in almost all transfer tasks.

This paper is organized as follows. Section 2 reviews related works including CNN for fault diagnosis and transfer learning. Section 3 proposes our intelligent fault diagnosis framework by using the transfer learning method. Experiment results and comparison are given in Section 4. Finally, conclusion and future work are drawn in Section 5.

The following notations will be used throughout this work: the symbol $\mathbb{R}$ is the real number set, and the symbol $\mathbb{Z}$ is the positive integer set. $(\cdot)^s$ and $(\cdot)^t$ represent the source and target domain information respectively.

2. Related works

In this section, some related work on intelligent fault diagnosis as well as CNN architecture are provided, and followed by a brief introduction associated with transfer learning and Wasserstein distance.

2.1. Convolutional neural networks

As the most well-known model in deep learning, in recent years, CNN dominates the recognition and detection problems in computer vision domain. The initial CNN architecture was proposed by LeCun et al. in works [22,23], which was inspired by Wiesel and Hubel's research works in cat recognition [24]. Main characteristics of CNN are local connections, shared weights, and local pooling [25]. The first two characteristics indicate the CNN model requires fewer parameters to detect local information of visual patterns than multilayer perceptron, while the last characteristic offers shift invariance to the network. Typically, 1-D CNN will be employed to this work to solve the bearing fault diagnosis problem, which has been widely used with great success in the study of speech recognition and document reading tasks.

In this work, a 1-D CNN model, as a base model, will be pre-trained in the source domain. The CNN extracts and learns characteristics of the task by stacking a series of layers with repeated components, including convolutional layers (with activation function), pooling layers, and fully connected layers (with an output classification layer) [26]. A typical CNN architecture is fed to a 1-D input layer to accept source domain signal, convolutional layers with rectified linear unit (ReLU) activation functions are followed for feature extraction, max-pooling layers are used to down-sample data size, and a fully connected layer combined with a softmax function is finally connected for classification (with pre-defined labels). To minimize the loss function, model parameters are tuned using Backpropagation algorithm [27] based on *Adam* optimizer, until the predefined maximum number of iterations is reached.

2.2. Transfer learning

Transfer learning can be a novel tool to solve the basic problem of unlabeled and insufficient data under diverse operating conditions in the target domain of mechanical systems, by utilizing the knowledge from source domain to improve the target domain learning performance. Some notations and definitions of transfer learning used in this work are first presented.

To begin with, we define a domain and a task respectively. Given a domain $\mathcal{D}$ in transfer learning defined as $\mathcal{D}=\{\mathcal{X},\mathbb{P}(X)\}$, where $\mathbb{P}(X)$ represents a marginal probability distribution of a feature space $\mathcal{X}$. Given predefined source and target domain datasets X^s and X^t, we have $X^s, X^t\in\mathcal{X}$. If $X^s\neq X^s$ and/or $\mathbb{P}(X^s)\neq\mathbb{P}(X^t)$, two domains $\mathcal{D}^s$ and $\mathcal{D}^t$ are with different distribution.

In the meantime, a task $\mathcal{T}$ in transfer learning is defined as $\mathcal{T}=\{\mathcal{Y},r(X)\}$, where $\mathcal{Y}$ represents a label space and $r(X)$ is a predictive function and $r(X)=\mathbb{P}(Y|X)$ is a conditional probability function. Since the classification categories are the same, source and target domains have the same label space, $\mathcal{Y}^s=\mathcal{Y}^t$. Then, we give the definition of transfer learning.

Definition 1. (*Transfer learning*) Transfer learning is proposed with the aim to learn a prediction function $r(X):X\longrightarrow Y$ for a learning task $\mathcal{T}^t$ by leveraging knowledge from source domain $\mathcal{D}^s$ and $\mathcal{T}^s$, where $\mathcal{D}^s\neq\mathcal{D}^t$ or $\mathcal{T}^s\neq\mathcal{T}^t$. In most of the cases, $\mathcal{D}^s$ contains a much larger dataset than $\mathcal{D}^t$ (i.e., the cardinality of $\mathcal{D}^s$ is larger than that of $\mathcal{D}^t$).

2.3. Wasserstein distance

Wasserstein distance is recently proposed by researchers [18] to tackle the training difficulty of generative adversarial networks (GAN) when facing discontinuous mapping problem of other distances and divergences in the generator, such as Total Variation (TV) distance and Kullback–Leibler (KL) divergence. As a promising way to measure the distance between two distributions for GAN training, Wasserstein distance could be applied to DTL for domain adaptation.

Given a compact metric set $\mathcal{H}, Prob(\mathcal{H})$ represents the space of probability measures on set $\mathcal{H}$. Wasserstein-1 distance (also called $Earth-Mover$ distance) is defined between two distributions $\mathbb{P}^s,\mathbb{P}^t\in Prob(\mathcal{H})$:

$$W\left(\mathbb{P}^s,\mathbb{P}^t\right)=\inf_{\mu\in\Pi\left(\mathbb{P}^s,\mathbb{P}^t\right)}\mathbb{E}_{\left(h^s,h^t\right)\sim\mu}\left[\|h^s-h^t\|\right]\tag{1}$$

where μ is a joint probability distribution and $\Pi\left(\mathbb{P}^s,\mathbb{P}^t\right)$ denotes the set $\mathcal{H}\times\mathcal{H}$ of all joint distributions $\mu\left(h^s,h^t\right)$ whose margins are $\mathbb{P}^s$ and $\mathbb{P}^t$ respectively. Wasserstein-1 distance can be viewed as an optimal transport problem, it aims to find an optimal transport plan $\mu\left(h^s,h^t\right)$. Intuitively, $\mu\left(h^s,h^t\right)$ indicates how much of 'mass' randomly transported from one place h^s over the domain of h^t, with the aim of transporting the distribution $\mathbb{P}^s$ into the distribution $\mathbb{P}^t$. Hence, Wasserstein-1 distance is the optimal transport plan with the lowest transport cost.

3. Wasserstein distance based deep transfer learning (WD-DTL)

3.1. Problem formulation

Since it is difficult to retrofit enough sensors in packaged equipment and industry labeling is often expensive, the challenge of domain adaptation is that there is no or limited labeled high-quality data can be collected in real industrial applications. For this reason, supervised domain adaptation approach by fine-tuning the pre-trained architecture to fit the new classification problem is not feasible. To solve this problem, many existing domain adaptation frameworks [17,15] use MMD to learn the invariant domain representations, which minimizes the target loss by the source loss with an additional maximum mean discrepancy metric. Our proposed approach WD-DTL is a promising alternative for domain adaptation by using the Wasserstein distance, which has been demonstrated with gradient superiority than MMD [18], to minimize the distributions between source domain and target domain. Although Wasserstein distance with MLP has been seen in few domain adaptation works in image classification tasks, to date there is no attempt to adopt this technique into industry or manufacturing and there is no attempt to enhance this technique in deep neural networks. It also has to be noted that we propose to use the CNN architecture to generate features for measuring the Wasserstein distance in both domains. The excellent local feature detection ability of CNN in manufacturing has been explored in work [28]. The problem with this work is formulated as follow:

The DTL with domain adaptation for fault diagnosis is an unsupervised problem, thus, we first define a source domain dataset with labels $Y^s=\{y_i^s\}_{i=1}^{N^s}$ of sample $X^s=\{x_i^s\}_{i=1}^{N^s}$, where $N^s\in\mathbb{R}$ number of samples in the source domain $\mathcal{D}^s$. In the meantime, an unlabeled target domain dataset $X^t=\{x_i^t\}_{i=1}^{N^t}$ is defined in the target domain $\mathcal{D}^t$. In most cases, source domain samples are sufficient enough to learn an accurate CNN classifier and with much larger data size than the target domain, which means $N^s\gg N^t$. It is also noted that data in source and target domains share the same feature space $(X^s, X^t\in\mathcal{X})$ but with different marginal distributions $\left(\mathbb{P}(X^s)\neq\mathbb{P}\left(X^t\right)\right)$.

The objective of this work is to construct a transferable framework, named WD-DTL, for the target task $\mathcal{T}^t$ to minimize target classification error $E^t\%=\Pr_{(x^t,y^t)\sim\mathcal{D}^t}\left[r\left(X^t\right)\neq y^t\right]$, with the help of the knowledge from source domain task $\mathcal{T}^s$ and the implementation of Wasserstein distance for domain adaptation.

The algorithm of WD-DTL will be trained by three iterative steps to achieve unsupervised or supervised diagnosis: a CNN-based feature extractor will be pre-trained with the source domain labeled dataset in Section 3.2; domain adaptation using Wasserstein distance between two different feature distributions through adversarial training by employing a domain critic will be explained in Section 3.3; and finally a classifier for classification is presented in Section 3.4.

3.2. CNN based feature extractor

First of all, we propose to use CNN to train the domain data. A CNN model is pre-trained with source domain labeled dataset X^s:

Convolution layer involves a filter $w\in\mathbb{R}^k$ and a bias $b\in\mathbb{R}$, which are applied to a filter size of k for calculating a new feature. An output feature v_i is obtained through the filter w and a non-linear *activation function* Γ with the following expression:

$$v_i=\Gamma\left(w*u_j+b\right)\tag{2}$$

where $u_j\in\mathbb{R}^{1\times k}$ is the input data representing j-th sub-vector of the source domain dataset X^s. '$*$' denotes the convolution operation. The non-linear activation function, such as hyperbolic tangent (tanh) or rectified linear unit (ReLu), is applied to reduce the risk of vanishing gradient which may impact the convergence of the optimization. Hence, the *feature map* is defined as $\mathbf{v}=[v_1,v_2,\ldots,v_L]$, where $L=(pN-s)/I_{cv}+1$ is the number of features and $I_{cv}\in\mathbb{Z}$ is the stride for convolution.

Max pooling layer is then applied over the feature map to extract the maximum feature values $\widehat{v}_i = \max_{\gamma=1,\dots,\beta} v_{\gamma+(i-1)l_{pl}}$ corresponding to its filter size β and the stride size l_{pl} for max pooling. The idea is to capture the maximum features over disjoint regions. Consequently, the features within the small window are similar and therefore illustrating the most important property of CNN.

By stacking multiple layers described above (with varying filter size), a multi-layer structure is constructed for feature description. The output features of the multi-layer structure are flattened and pass to fully-connected layers for classification, resulting in probability-distributed final outputs $\tilde{y}_i^s$ over labels. For the pre-trained CNN in the source domain, *Softmax* function [29] is selected for classification over the final feature map.

To compute the difference between the predicted label, $\tilde{y}_i^s$, and the ground truth, y_i^s, in the source domain, cross-entropy function l_c is used to compute the loss:

$$l_c = \frac{1}{N^s}\sum_{i=1}^{N^s} -y_i^s \log \tilde{y}_i^s - (1-y_i^s)\log(1-\tilde{y}_i^s). \quad (3)$$

3.3. Domain adaptation via Wasserstein distance

The next problem is to solve the distribution difference between the source and target datasets. To tackle this problem, we utilize Wasserstein-1 distance to learn invariant feature representations in a common latent space between two different feature distributions through adversarial training.

The network structure before fully-connected layer of pre-trained CNN model is used as the feature extractor to learn the invariant feature representations from both domains. Given two mini-batch of instances $\{x^s\}_{i=1}^n$ and $\{x^t\}_{i=1}^n$ from X^s and X^t for $n < N^s$ and N^t. Both instances are passed through a parameter function $r_f : \mathcal{X} \rightarrow \mathcal{H}$ (i.e., feature extractor) with corresponding network parameter θ_f that directly generate source features $h^s = r_f(x^s)$ and target features $h^t = r_f(x^t)$. Let $\mathbb{P}^s$ and $\mathbb{P}^t$ be the distribution of h^s and h^t respectively.

The aim of domain adaptation via Wasserstein distance [18] is to optimize the parameter θ_f to reduce the distance between distributions $\mathbb{P}^s$ and $\mathbb{P}^t$. We introduce a domain alignment critic to learn a solution $r_c : \mathcal{H} \rightarrow \mathbb{R}$ that maps the source and target features to a real number, with corresponding parameters θ_c. However, the *infimum* in Eq. (1) is highly intractable to handle directly. Thanks to the Kantorovich-Rubinstein duality [30], the Wasserstein-1 distance can be computed by

$$W(\mathbb{P}^s,\mathbb{P}^t) \triangleq \sup_{\|r_c\|\leqslant 1} \mathbb{E}_{h^s\sim\mathbb{P}^s}\left[r_c(h^s)\right] - \mathbb{E}_{h^s\sim\mathbb{P}^t}\left[r_c(h^s)\right] \quad (4)$$

where the *supremum* is over all the 1-Lipschitz functions $r_c : \mathcal{H} \rightarrow \mathbb{R}$. The empirical Wasserstein-1 distance can be approximately computed as follow:

$$l_{wd} = \frac{1}{N^s}\sum_{x^s\in X^s} r_c(r_f(x^s)) - \frac{1}{N^t}\sum_{x^t\in X^t} r_c(r_f(x^t)) \quad (5)$$

where l_{wd} denotes the domain alignment critic loss between the source data X^s and the target data X^t

We now comes to the optimization problem that finds the maximum of Eq. (5) while enforcing the Lipschitz constraint. Arjovsky et al. [18] proposed a weight clipping method after each gradient update to force the parameters θ_c inside a compact space. However, this method is time consuming when clipping parameter is large and might result in vanishing gradients when the number of layers is set too big. To solve this problem, distribution $\mathbb{P}^r$ is defined along straight lines between pairs of points from the source and target distribution $\mathbb{P}^s$ and $\mathbb{P}^t$ [20,19] and a gradient penalty $l_{grad} = (\|\nabla_{\mathbf{h}} r_c(\mathbf{h})\|_2 - 1)^2$ is introduced to train the domain alignment critic with respects to parameters θ_c, where the feature representations $\mathbf{h} \in \mathbb{P}^r$ consist of the generated source and target domain features (i.e., h^s and h^t), as well as points h^r which are randomly selected along the straight line between h^s and h^t pairs.

Due to the fact that the Wasserstein-1 distance is differentiable and continuous almost everywhere, we here to train the critic till optimally by solving the following optimization problem:

$$\max_{\theta_c}\{l_{wd} - \rho l_{grad}\} \quad (6)$$

where ρ is the balancing coefficient.

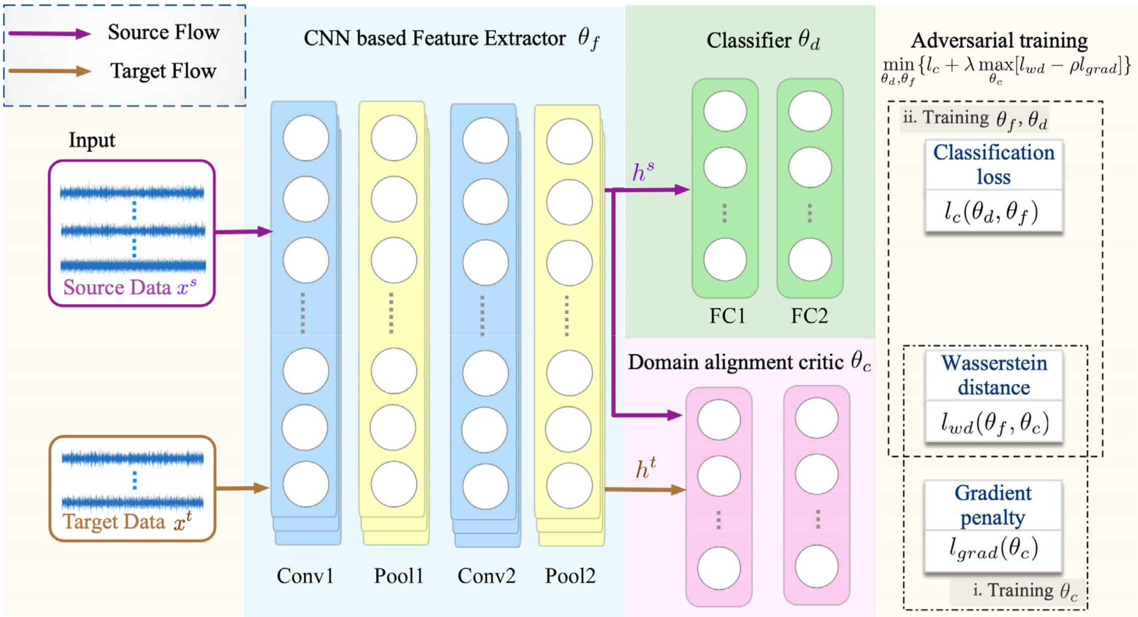

Source Flow
Target Flow
CNN based Feature Extractor θ_f
Classifier θ_d
Adversarial training
$\min_{\theta_d,\theta_f}\{l_c + \lambda \max_{\theta_c}[l_{wd} - \rho l_{grad}]\}$
Input
ii. Training θ_f, θ_d
Classification loss
$l_c(\theta_d, \theta_f)$
h^s
Source Data x^s
FC1
FC2
Domain alignment critic θ_c
Wasserstein distance
$l_{wd}(\theta_f, \theta_c)$
h^t
Target Data x^t
Gradient penalty
$l_{grad}(\theta_c)$
Conv1
Pool1
Conv2
Pool2
i. Training θ_c

Fig. 1. WD-DTL framework of the fault diagnosis, which is comprised of three sub networks: a CNN based feature extractor, a domain alignment critic for learning feature representations via Wasserstein distance, and a classifier for classification. The two-stage adversarial training process is also illustrated.

3.4. Classification with classifier

The above Section 3.3 proposed an unsupervised feature learning for domain adaptation, which may cause the learned feature representations in both domains are not discriminative enough. As stated in Section 3.1, our final objective is to develop an accurate classifier, WD-DTL, for target domain $\mathcal{D}^t$, which requires to incorporate the labeled supervised learning of source domain data (and target domain if available) into the invariant feature learning problem. A classifier [31] (with two fully-connected layers) is then employed into the representation learning approaches to further reduce the distance between source and target feature distributions. In this step, parameters of domain alignment critic θ_c are the ones trained in Section 3.3, while the parameters θ_f will be modified to optimize the minimum operator.

Now the final objective function of the classification loss can be expressed in terms of the cross-entropy loss l_c of the classifier according to Eq. (3) and the empirical Wasserstein distance l_{wd} which associated with domain discrepancy, i.e:

$$\min_{\theta_d,\theta_f}\left\{l_c+\lambda\max_{\theta_c}\left[l_{wd}-\rho l_{grad}\right]\right\} \tag{7}$$

where θ_d denotes the parameters for the classifier and λ is the hyper-parameter that determines the extent of domain confusion. We omit the gradient penalty l_{grad} (i.e., $\rho=0$) when optimizing the minimum operator as it should not affect the representation learning process.

3.5. WD-DTL approach

Hence, the overall framework of intelligent fault diagnosis approach in this work is illustrated in Fig. 1 and a detailed algorithm is summarized in Algorithm 1.

Algorithm 1 Training procedure of WD-DTL.

```
Require: source and target dataset: $X^s$ and $X^t$; the learning rate for domain alignment critic: $\alpha_1$; the learning rate for classifier and feature learning: $\alpha_2$; the batch size: $n$; critic training step: $C$; balance coefficients: $\rho$ and $\lambda$.
Require: initial CNN based feature extractor parameters: $\theta_f$; initial domain alignment critic parameters: $\theta_c$; initial classifier parameters: $\theta_d$.
while $\theta_f, \theta_c$, and $\theta_d$ has not converged do
    Sample $\{x_i^s, y_i^s\}_{i=1}^n$, a batch from source dataset $X^s$.
    Sample $\{x_i^t\}_{i=1}^n$, a batch from target dataset $X^t$.
    for $i = 0,\ldots, C$
        $h^s \leftarrow r_f(x^s)$ , $h^t \leftarrow r_f(x^t)$
        $\mathbf{h} \leftarrow \left\{h^s, h^t, h^r\right\}$
        $l_{grad} \leftarrow (\|\nabla_{\mathbf{h}} r_c(\mathbf{h})\|_2 - 1)^2$
        $\theta_c \leftarrow \theta_c + \alpha_1 \nabla_{\theta_c}\left[l_{wd}(x^s, x^t) - \rho l_{grad}(\mathbf{h})\right]$
    end for
    $\theta_d \leftarrow \theta_d - \alpha_2 \nabla_{\theta_d} l_c(x^s, y^s)$
    $\theta_f \leftarrow \theta_f - \alpha_2 \nabla_{\theta_f}[l_c(x^s, y^s) + \lambda l_{wd}(x^s, x^t)]$
end while
```

3.6. Gradient of the WD

Optimization algorithm is used to train the WD-DTL model to minimize Eq. (7). When adopting the domain classifier, other distances such as KL, JS and TV losses will cause gradient vanishing problem on low dimensional manifolds. On the other hand, Theorem 1 in [18] proved that WD is a much more sensible cost function for the training of neural networks as it is continuous everywhere, and differentiable almost everywhere. In this work, according to Eq. (5), we could calculate the gradient of l_{wd} with respect to θ_f of one instance x by chain rule

$$\frac{\partial l_{wd}}{\partial \theta_f}=\frac{\partial l_{wd}}{\partial r_c}\frac{\partial r_c}{\partial r_f}\frac{\partial r_f}{\partial \theta_f}. \tag{8}$$

For an instance either in the source domain where $x^s\sim\mathbb{P}(X^s)$ or in the target domain where $x^t\sim\mathbb{P}(X^t)$, we have that

$$\frac{\partial l_{wd}}{\partial \theta_f}=\frac{\partial r_c}{\partial r_f}\frac{\partial r_f}{\partial \theta_f} \tag{9}$$

or

$$\frac{\partial l_{wd}}{\partial \theta_f}=-\frac{\partial r_c}{\partial r_f}\frac{\partial r_f}{\partial \theta_f} \tag{10}$$

respectively. Therefore, stable gradients will be provided by the WD wherever the instance is.

3.7. Compared with MMD

MMD is the most common seen distance measure used in existing transfer networks [32,17,15]. The MMD metric is a special case of IPMs which measures the distance between two probability distributions via mapping the samples into a Reproducing Kernel Hilbert Space (denote by $\mathcal{H}_k$) associated with a given kernel $k:\mathcal{X}\times\mathcal{X}\rightarrow\mathbb{R}$. Therefore, via the kernel trick, the structure of MMD is simpler than WD in where the domain alignment critic network is not required to approximately maximum Eq. (6). The squared formulation of MMD l_{MMD} is that

$$l_{MMD}^2\triangleq\|\mathbb{E}_{x^s\sim\mathbb{P}(X^s)}[\phi(x^s)]-\mathbb{E}_{x^t\sim\mathbb{P}(X^t)}\left[\phi(x^t)\right]\|_{\mathcal{H}_k}^2 \tag{11}$$

where $\phi(x)$ representing the features extracted before classification. However, compared to Eq. (5) of WD, it implies that the computational cost of MMD will increase quadratically with the sample number, which limits the applicability of MMD in many real life applications with large data sets. This is another reason we use WD as distance measure in our network.

4. Experiments

4.1. Data description

To validate the effectiveness of the proposed DTL method for fault diagnosis problem, we introduce a benchmark bearing fault dataset acquired by Case Western Reserve University (CWRU) data center. An experiment test-bed (see Fig. 2) is used to conduct the signals for the detection of defects on bearings. Four types of bearing conditions are inspected, namely health condition, fault on inner races, fault on outer races, and fault on rollers. All those situations are sampled with 12 kHz frequency. Meanwhile, each fault type is running with different levels of fault severity (0.007-inch, 0.014-inch, and 0.021-inch fault diameters). Each type of faulted bearing was equipped with the test motor, which runs under four different motor speeds (i.e., 1797 rpm, 1772 rpm, 1750 rpm, and 1730 rpm). Vibration signal of each experiment was recorded for fault diagnosis.

Data pre-processing: Simple data pre-processing techniques are applied to the bearing datasets:

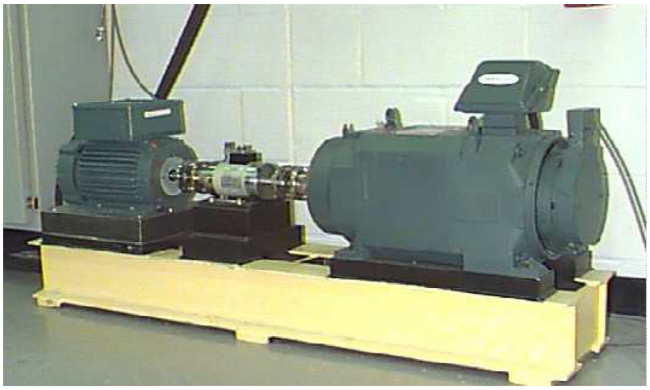


Fig. 2. Experimental test-bed in Case Western Reserve University (CWRU) for bearing fault diagnosis.

1. To modify the faulty signal to a stationary process, we here divide the samples to keep each sample has 2000 measurements in both $\mathcal{D}^s$ and $\mathcal{D}^t$.
2. Fast Fourier transform (FFT) computes the power spectrum in the frequency domain of every sample.
3. Clip the left side of the power spectrum calculated by FFT as the input for WD-DTL. Therefore, each input sample has 1000 measurements.

We proposed three transfer scenarios, including two unsupervised scenarios and one supervised scenario (refer to Table 1), they are:

1. Unsupervised transfer between motor speeds **(US-Speed)**: For this scenario, we test the data with 12 kHz sampling frequency acquired at the drive end of the motor, and ignore the level of fault severities. Thus, we construct 4-way classification tasks (i.e., health condition, and three fault conditions with faults on inner race, outer race and roller), across 4 domains with different motor speeds: 1797 rpm ($\mathbf{US}(\mathbf{A})$), 1772 rpm ($\mathbf{US}(\mathbf{B})$), 1750 rpm ($\mathbf{US}(\mathbf{C})$), and 1730 rpm ($\mathbf{US}(\mathbf{D})$). In total, for this scenario, we evaluate our proposed method over 12 transfer tasks.
2. Unsupervised transfer between datasets at two sensor locations **(US-Location)**: For this scenario, we focus on domain adaptation between different sensor locations but ignore the level of fault severities and the differences in motor speeds. Again, we construct 4-way classification tasks for health and three fault conditions, across 2 domains (2 tasks) where vibration acceleration data acquired by two sensors placed at the drive end ($\mathbf{US}(\mathbf{E})$) and fan end ($\mathbf{US}(\mathbf{F})$) of the motor housing respectively.
3. Supervised transfer between datasets at two sensor locations **(S-Location)**: this scenario uses the same settings as the previous scenario **US-Location**, except for the specified change of adding a small amount of labeled data ($\sim 0.5\%$) of target domain in source domain which aims to enhance the classification performance.

To evaluate the efficiency of our proposed approach WD-DTL on bearing fault diagnosis problem, other approaches are also tested on the same dataset for comparison purpose:

- *CNN (no transfer):* This model is the pre-trained network described in Section 3.2, which is trained based on the labeled source data and applied to test the classification result on the target domain directly.
- *DAN:* We follow the idea in work [15], which proposed a deep adaptation network (DAN) for learning transferable features via MMD in deep neural networks.
- In addition, to evaluate the feature extraction ability of CNN compared to the use of conventional statistical features, results of traditional transfer learning methods using statistical (handcrafted) features [33], including transfer component analysis *(TCA)* [34], joint distribution adaptation *(JDA)* [35], and CORrelation ALignment *(CORAL)* [36], are also provided for comparison.

This work will mainly focus on the comparison between those deep transfer learning methods (DAN and WD-DTL) and CNN.

4.2. Implementation details

Tensorflow [37] is used as the software framework for all our experiments using deep learning flow, and those models are all trained with *Adam* optimizer. We test each approach for five times over 5000 iterations and record the best result of each test. We take the averages and 95% confidential interval of classification accuracy for comparison. The sample size for motor speed tasks $(\mathbf{A}), (\mathbf{B}), (\mathbf{C})$, and $(\mathbf{D})$ are 1026, 1145, 1390, and 1149 respectively. The sample size for different sensor location tasks $(\mathbf{E})$ and $(\mathbf{F})$ are 3790 and 4710 respectively. The batch size n is fixed be fixed at 32 for all experiments.

CNN: Our CNN architecture is comprised of two convolutional layers ($Conv1 - Conv2$), two max-pooling layers ($Pool1 - Pool2$), and two fully-connected layers ($FC1 - FC2$). The activation function in the output layer is *Softmax* while *ReLu* is used in convolutional layers. The neuron number in $FC1$ and $FC2$ are 128 and 4, respectively. Filters, kernel size, and stride of each layer can refer to Table 2. To achieve a fair comparison cross different methods, we fine-tune the CNN base models which achieve their best validation accuracies before transfer.

DAN: The convolutional layers ($Conv1 - Conv2$) of the CNN network is used to be the feature extractor. Then, to minimize the domain distance between the source and target domains, FC1 is used as the hidden layer for adaptation. The final representations of the hidden layer in both domains are embedded in RKHS to reduce the MK-MMD distance. The final objective function is the combination of the MK-MMD loss and the classification loss. The parameter settings of MMD can refer to [15], and best classification accuracies are obtained for transfer scenarios by tuning the balancing coefficient for the discrepancy loss.

$\mathbf{WD - DTL}$: WD-DTL method has been summarized in Fig. 1 and Algorithm 1. Similar to DAN, convolutional layers (Conv1-Conv2) are used to extract features. The nodes of hidden layers in the domain alignment critic network are set to 128 and 1, respectively. The training step C is set to 10. The learning rates for the classifier and the domain alignment critic are $\alpha_1 = 10^{-3}$ and $\alpha_2 = 2 \times 10^{-4}$ respectively. The gradient penalty ρ is set to 10. Balance coefficient λ for optimizing the minimum operator is 0.1 and 0.8 for motor speed transfer and sensor location transfer, respectively.

In terms of the traditional transfer learning methods TCA, JDA and CORAL, the regularization term λ is chosen from {0.001 0.01 0.1 1.0 10 100}. SVM is used in TCA and CORAL for classification.

4.3. Results and discussion

The results of transfer tasks for WD-DTL, DAN, CNN, and other conventional approaches are compared in Table 3. For the transfer task with unlabeled data set in target domain (i.e., scenario **US-Speed** and **US-Location**), we can observe that WD-DTL significantly outperforms CNN with a large margin, which achieves

Table 1
Summaries of transfer scenarios and tasks.

Scenario	Unsupervised or Supervised	Transfer task	Transfer condition	Training	Testing	Classification
US-Speed	Unsupervised	**US(A)-US(B)**	From: 1797 rpm, To:1772 rpm	Source labeled: 100% arget unlabeled: 100%	Target unlabeled: 100%	Normal (denoted by 0) Inner Race (denoted by 1) Outer Race (denoted by 2) Roller (denoted by 3)
		US(A)-US(C)	From: 1797 rpm, To:1750 rpm			
		US(A)-US(D)	From: 1797 rpm, To:1730 rpm			
		US(B)-US(A)	From:1772 rpm, To:1797 rpm			
		US(B)-US(C)	From:1772 rpm, To:1750 rpm			
		US(B)-US(D)	From:1772 rpm, To:1730 rpm			
		US(C)-US(A)	From:1750 rpm, To:1797 rpm			
		US(C)-US(B)	From:1750 rpm, To:1772 rpm			
		US(C)-US(D)	From:1750 rpm, To:1730 rpm			
		US(D)-US(A)	From:1730 rpm, To:1797 rpm			
		US(D)-US(B)	From:1730 rpm, To:1772 rpm			
		US(D)-US(C)	From:1730 rpm, To:1750 rpm			
US-Location	Unsupervised	**US(E)-US(F)**	From:Drive End, To:Fan End			
		US(F)-US(E)	From:Fan End, To:Drive End			
S-Location	Supervised	**S(E)-S(F)**	From:Drive End, To:Fan End	Source labeled: 100% Target labeled:>0.5%	Target unlabeled: 25%	
		S(F)-S(E)	From:Fan End, To:Drive End			

approximately 13.6% and 25% increases in average accuracies for motor speed and sensor location transfer tasks, respectively. In addition, the WD-DTL transfer accuracies are better than most of the DAN results (average 5% increase), except transfer task **US**(**D**) → **US**(**A**) which results in less than 1% accuracy difference. Furthermore, the WD-DTL significantly increases diagnosis accuracies of both supervised and unsupervised tasks, when compared to conventional transfer learning methods TCA, JDA, and CORAL, except the unsupervised task **US**(**F**) → **US**(**E**) for JDA.

To summarize the results, we can make the following observations: 1) WD-DTL achieves the best transfer accuracies with 95.75% average score, confirming the effectiveness of Wasserstein distance in learning transferable features using CNN-based model; 2) Without domain adaptation, CNN method already has the ability to achieve good classification performance for the motor speed transfer tasks, due to its excellent feature detection ability; 3) The accuracies of CNN, DAN and WD-DTL on transfer tasks of scenario **US-Location** are not better than the transfer tasks of scenario **US-Speed**, due to the characteristics of signals obtained at different sensor location (Fan End and Drive End) are more different than the difference between motor speeds; and 4) The proposed WD-DTL approach shows a good ability to solve supervised problem with a small number of labeled data. Supervised transfer tasks **S(E)**→ **S(F)** and **S(E)** → **S(F)** are carried out using only 0.5% sample size of the unsupervised case, but achieve as good as performance compared to the unsupervised case while using 100% unlabeled sample. Further analysis of the effect of sample size for both supervised and unsupervised transfer learning will be shown in Section 4.4.2.

4.4. Empirical analysis

4.4.1. Feature visualization

To further evaluate the transfer performance of the proposed WD-DTL framework, t-distributed stochastic neighbor embedding (t-SNE) is employed to perform the nonlinear dimensionality reduction for network visualization. For comparison purpose, CNN and DAN transfer results for same tasks are also provided.

For transfer tasks between motor speeds, i.e., scenario **US-Speed**, we randomly choose task **US**(**C**) → **US**(**A**) to visualize the learned feature representations under different motor speeds. Fig. 3 shows the comparison results. It can be observed that the clusters in Fig. 3(c) formed by our proposed WD-DTL are better separated than the CNN network result in Fig. 3(a) (no transfer case) and the DAN domain adaptation result in Fig. 3(b). For example, in Fig. 3(a) with CNN approach, three types of fault features are inspected with large overlapped areas, and some outer-race faults (yellow color with label 2) fall into other fault types. Similarly, in Fig. 3(b) with DAN approach, outer-race faults is also hardly be separated from other fault types. With our WD-DTL approach, four conditions are clearly separated into different clusters. More importantly, we can observe the obvious improvement of domain adaptation due to the source and target domain features are almost mixed into the same cluster.

Table 2
Parameters in the CNN model.

Layer	Filters	Kernel size	Stride
Conv1	8	1 × 20	2
Pool1	–	1 × 2	2
Conv2	16	1 × 20	2
Pool2	–	1 × 2	2

Table 3
Performance of transfer tasks (Accuracy %).

	TCA	JDA	CORAL	CNN	DAN	WD-DTL
US(A)→US(B)	26.55	65.07 (± 7.55)	59.18	82.75 (± 6.77)	92.97 (± 3.88)	**97.52** (± 3.09)
US(A)→US(C)	46.80	51.31 (± 1.56)	62.14	78.65 (± 4.54)	85.32 (± 5.26)	**94.43** (± 2.99)
US(A)→US(D)	26.57	57.70 (± 8.59)	49.83	82.99 (± 5.89)	89.39 (± 4.37)	**95.05** (± 2.12)
US(B)→US(A)	26.63	71.19 (± 1.21)	53.57	84.14 (± 6.63)	94.43 (± 2.95)	**96.80** (± 1.10)
US(B)→US(C)	26.60	69.80 (± 5.67)	57.28	85.41 (± 9.44)	90.43 (± 4.62)	**99.69** (± 0.59)
US(B)→US(D)	26.57	88.50 (± 1.96)	60.53	86.09 (± 4.63)	87.37 (± 5.42)	**95.51** (± 2.52)
US(C)→US(A)	26.63	56.42 (± 2.52)	54.03	76.50 (± 3.76)	89.88 (± 1.57)	**92.16** (± 2.61)
US(C)→US(B)	26.66	69.18 (± 1.90)	76.66	82.75 (± 5.51)	92.93 (± 1.57)	**96.03** (± 6.27)
US(C)→US(D)	46.75	77.45 (± 0.83)	70.34	87.04 (± 6.81)	90.66 (± 5.24)	**97.56** (± 3.31)
US(D)→US(A)	46.74	61.72 (± 5.48)	59.78	79.23 (± 6.96)	**90.88** (± 1.82)	89.82 (± 2.41)
US(D)→US(B)	46.79	74.03 (± 0.86)	59.73	79.73 (± 5.49)	87.91 (± 2.42)	**95.16** (± 3.67)
US(D)→US(C)	26.60	65.24 (± 4.18)	63.02	80.64 (± 4.23)	92.94 (± 3.96)	**99.62** (± 0.80)
Average	33.32	67.35 (± 3.53)	56.01	82.10 (± 5.89)	90.42 (± 3.59)	**95.75** (± 2.62)
US(E)→US(F)	19.05	57.35 (± 0.47)	47.97	39.07 (± 2.22)	56.89 (± 2.73)	**64.17** (± 7.16)
US(F)→US(E)	20.45	**66.34** (± 4.47)	39.87	39.95 (± 3.84)	55.97 (± 3.17)	64.24 (± 3.87)
Average	19.75	61.85 (± 2.47)	43.92	39.51 (± 3.03)	56.43 (± 2.95)	**64.20** (± 5.52)
S(E)→S(F)	20.43	65.48 (± 0.57)	51.77	54.04 (± 7.67)	59.68 (± 4.61)	**65.69** (± 3.74)
S(F)→S(E)	19.02	59.07 (± 0.56)	47.88	50.47 (± 5.74)	58.78 (± 5.67)	**64.15** (± 5.52)
Average	19.73	62.28 (± 0.57)	49.83	52.26 (± 6.71)	59.23 (± 5.14)	**64.92** (± 4.63)

For transfer tasks between different sensor locations, t-SNE results of transfer task $\mathbf{US(E)} \rightarrow \mathbf{US(F)}$ are shown in Fig. 4. It can be viewed that even WD-DTL shows better clustering result than CNN and DAN, faults types 1, 2, and 3 are hard to be separated clearly into individual clusters. It must be emphasized that the above results are carried out by using 100% (4710) sample size in the target domain, and even in this case, the performance is not satisfied enough. This raises the problem of how to enhance the transfer learning performance when signals in the source and target domains are relevant but not similar enough. We investigate this problem in the next subsection.

4.4.2. Effect of sample size on unsupervised and supervised accuracy

Next, we investigate the influence of data size on transfer task accuracy for our proposed method WD-DTL. For each sample number tested, same experiment is repeated five times and transfer learning accuracies are recorded. As it has known that our propose WD-DTL method already achieved very good performance (average 95.75% accuracy in Table 3) for unsupervised transfer scenario **US-Speed**. Considering other two scenarios in Table 1, Fig. 5 displays the accuracy variation curve for WD-DTL of tasks $\mathbf{US(E)} \rightarrow \mathbf{US(F)}$ and $\mathbf{S(E)} \rightarrow \mathbf{S(F)}$ with respect to scenario **US-Location** and **S-Location**. Diagnosis accuracies will be saturated around a fixed value when sample number larger than 2500, thus we only show the result from 10 to 2500.

In Fig. 5(a), it can be observed that the accuracy of WD-DTL is increased from 59.47% and the final test accuracy is confined around 64%. While the sample number is increasing, fault diagnosis accuracies of WD-DTL approach are all higher than DAN and CNN. This analysis reveals a limitation of the proposed WD-DTL that, for this unsupervised scenario with large discrepancies between domains, the improvement is limited (less than 5%) even with 100% sample number in the target domain. To solve this problem, in Fig. 5(b), we employ a small amount of labeled data to improve the fault diagnosis accuracy, which is associated with the case with limited labeled data in real industrial application. The plot shows that when the labeled sample size larger than 20 of 4710 the transfer learning accuracy of WD-DTL will surpass the case in Fig. 5(a) with 100% sample size (blue zone in Fig. 5 (a)). More specifically, only using 100 labeled sample, (equivalent to 25 for each fault categorization) could achieve 80% transfer learning accuracy, indicating our proposed WD-DTL is also an optimal framework for supervised transfer task.

Based on the above discussions, we hereby offer two solutions for manufacturers of using the proposed WD-DTL approach: 1) when facing the transfer tasks between similar signals in source and target domains, such as transfer learning between different motor speeds, unsupervised transfer learning with unlabeled data is enough to obtain very good fault diagnosis accuracy (higher than 95%); and 2) when facing the transfer tasks between relevant sig-

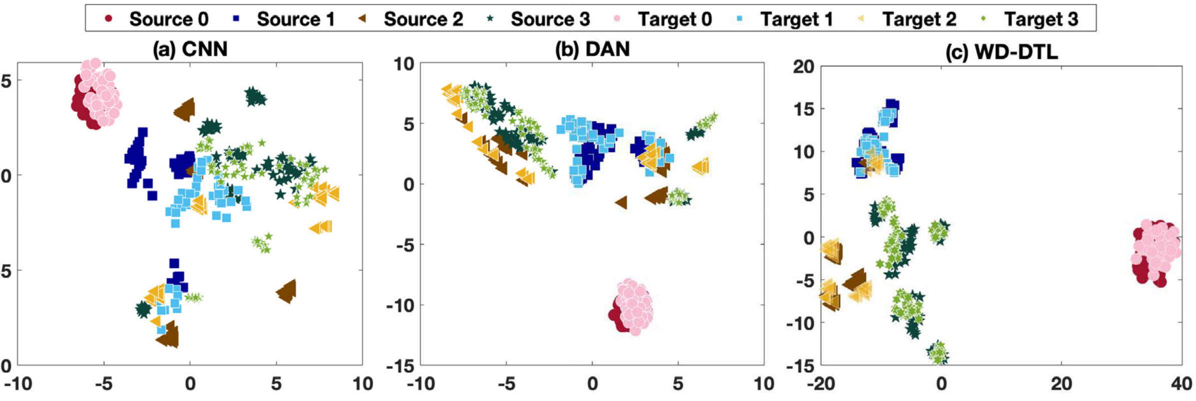


Fig. 3. Network visualization revealed by t-SNE embeddings of transfer task $\mathbf{US(C)} \rightarrow \mathbf{US(A)}$ with: (a) CNN approach, (b) DAN approach, and (c) WD-DTL approach. t-SNE is applied on the features in the last layer assigned by CNN-based feature extractor network, for both source and target domains. Four colors/shapes represent four conditions, namely normal condition, fault on inner race, fault on outer race, and fault on roller (with corresponding labels 0–3).

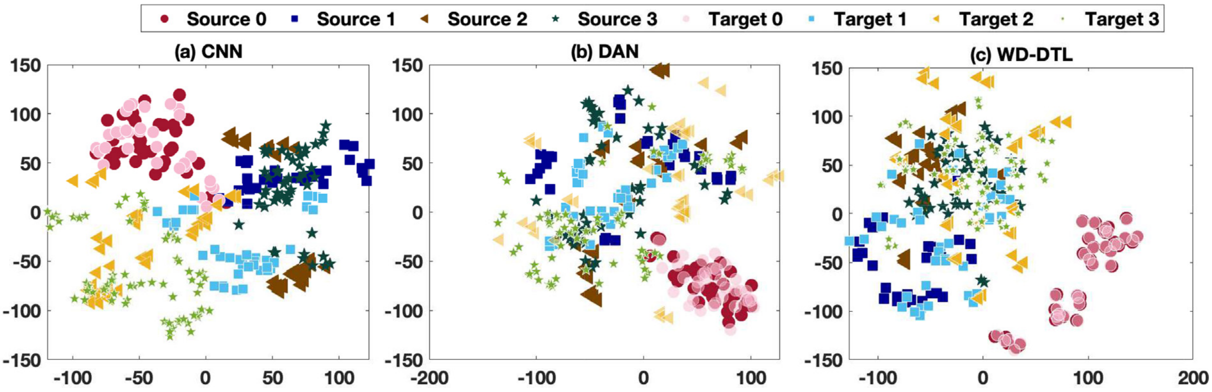


Fig. 4. Network visualization of transfer task **US**(**E**) → **US**(**F**) with: (a) CNN approach, (b) DAN approach, and (c) WD-DTL approach.

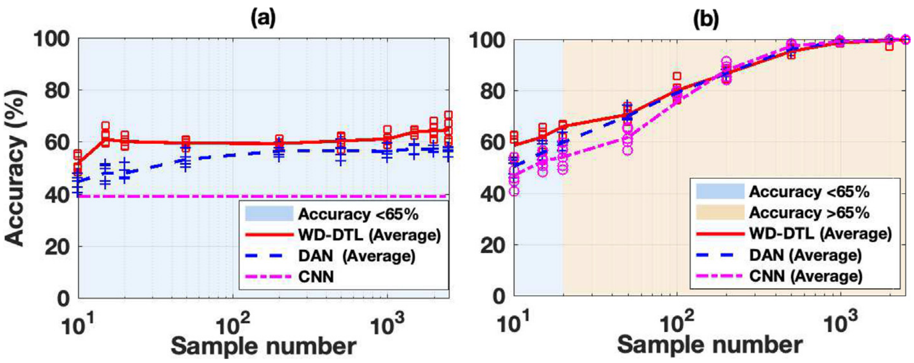


Fig. 5. Accuracy variation curve of task (a) **US**(**E**) → **US**(**F**) and (b) **S**(**E**) → **S**(**F**), where sample number is increased from 10 to 2500. Red, blue, and purple lines present the evolution of the average accuracy of five times' results under different sample number.

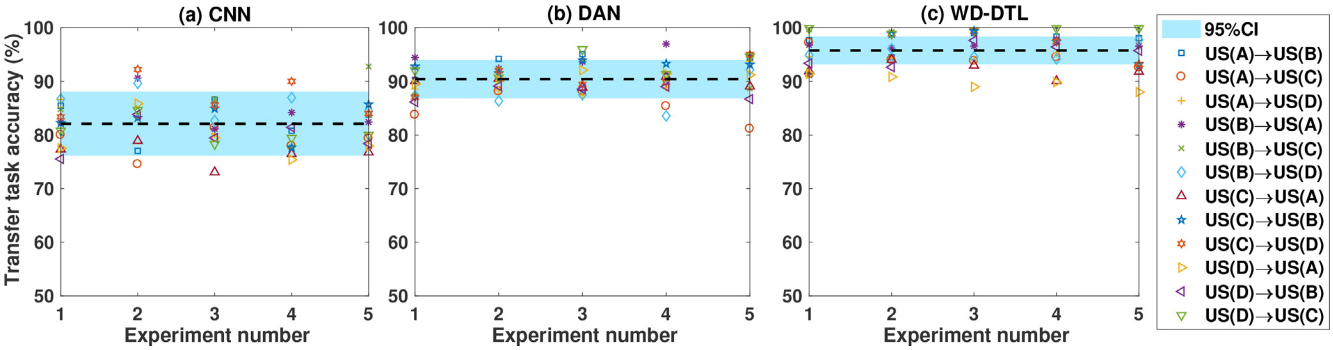


Fig. 6. Experiment number vs. transfer task accuracy of motor speed transfer tasks, for (a) CNN approach, (b) DAN approach, and (c) WD-DTL approach. Black dotted lines represent the average scores. Blue zone is the 95% confidential interval of each approach.

nals but not similar enough, such as transfer learning between different sensor locations, a small amount of labeled sample will greatly improve the transfer learning accuracy compared to the unsupervised case with a large amount of unlabeled sample data.

4.4.3. Algorithm robustness evaluation

The robustness of our proposed algorithm WD-DTL is investigated and compared with CNN and DAN approaches. We run each task for five times and store the transfer accuracy of each task. Fig. 6 gives an illustration of the variation of transfer task accuracy on 12 tasks of **US-Speed** scenario. We can observe that not only the WD-DTL accuracy is higher than other two approaches but also it has a narrower 95% confidential interval than other two approaches. This confirms our motivation of using CNN-based network and Wasserstein distance for domain adaptation, since both the accuracy and model robustness of feature transferability are enhanced by using our proposed algorithm.

5. Conclusion

To achieve intelligent fault diagnosis, we proposed a novel Deep Transfer Learning architecture via Wasserstein Distance (WD-DTL) to enhance the domain adaptation ability. WD-DTL is constructed based on a deep learning flow (CNN architecture) to extract features and introduces a domain alignment critic to learn domain invariant feature representations. Through an adversarial training process, WD-DTL significantly reduces the domain discrepancy thanks to its gradient property of Wasserstein distance over other state-of-the-art distances and divergences. Our proposed method is tested on a CRWU benchmark bearing fault diagnosis dataset and compared with the base CNN model, DAN metric and other traditional transfer learning methods over 16 transfer tasks. Performance of all the transfer tasks demonstrates that WD-DTL outperforms other approaches with much better classification accuracies. Empirical results also show that 1) our proposed method achieves higher robustness for motor speed transfer tasks, and 2) WD-DTL is a novel approach which could contribute to solving both unlabeled and insufficient labeled data problems in real industry applications. Future work includes investigating more transfer scenarios (e.g. transfer learning between different machines) for intelligent fault diagnosis and optimizing the architecture of our proposed algorithm.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Key R&D Program of China [Grant No. 2018YFB1701202], and in part by the National Natural Science Foundation of China [Grant No. 51905197].

References

[1] Y. Yuan, G. Ma, C. Cheng, B. Zhou, H. Zhao, H.-T. Zhang, H. Ding, A general end-to-end diagnosis framework for manufacturing systems, Natl. Sci. Rev. 7 (2) (2020) 418–429.

[2] D.-T. Hoang, H.-J. Kang, A survey on deep learning based bearing fault diagnosis, Neurocomputing 335 (2019) 327–335.

[3] W. Fan, Q. Zhou, J. Li, Z. Zhu, A wavelet-based statistical approach for monitoring and diagnosis of compound faults with application to rolling bearings, IEEE Trans. Autom. Sci. Eng. 15 (4) (2017) 1563–1572.

[4] B. Samanta, K. Al-Balushi, Artificial neural network based fault diagnostics of rolling element bearings using time-domain features, Mech. Syst. Signal Process. 17 (2) (2003) 317–328.

[5] P. Tamilselvan, P. Wang, Failure diagnosis using deep belief learning based health state classification, Reliab. Eng. Syst. Saf. 115 (2013) 124–135.

[6] X. Li, X.-D. Jia, W. Zhang, H. Ma, Z. Luo, X. Li, Intelligent cross-machine fault diagnosis approach with deep auto-encoder and domain adaptation, Neurocomputing (2019).

[7] G. Ma, Y. Zhang, C. Cheng, B. Zhou, P. Hu, Y. Yuan, Remaining useful life prediction of lithium-ion batteries based on false nearest neighbors and a hybrid neural network, Appl. Energy 253 (2019), 113626.

[8] K. Xu, S. Li, X. Jiang, Z. An, J. Wang, T. Yu, A renewable fusion fault diagnosis network for the variable speed conditions under unbalanced samples, Neurocomputing 379 (2020) 12–29.

[9] R. Gopalan, R. Li, R. Chellappa, Domain adaptation for object recognition: an unsupervised approach, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 999–1006. .

[10] V.M. Patel, R. Gopalan, R. Li, R. Chellappa, Visual domain adaptation: a survey of recent advances, IEEE Signal Process. Mag. 32 (3) (2015) 53–69.

[11] J. Deng, Z. Zhang, F. Eyben, B. Schuller, Autoencoder-based unsupervised domain adaptation for speech emotion recognition, IEEE Signal Process. Lett. 21 (9) (2014) 1068–1072.

[12] Y. Yao, G. Doretto, Boosting for transfer learning with multiple sources, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE conference on, IEEE, 2010, pp. 1855–1862. .

[13] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, in: Advances in Neural Information Processing Systems, 2014, pp. 3320–3328. .

[14] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, T. Darrell, Deep domain confusion: Maximizing for domain invariance, arXiv preprint arXiv:1412.3474 (2014). .

[15] M. Long, Y. Cao, J. Wang, M.I. Jordan, Learning transferable features with deep adaptation networks, in: Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37, JMLR. org, 2015, pp. 97–105.

[16] M. Long, H. Zhu, J. Wang, M.I. Jordan, Deep transfer learning with joint adaptation networks, in: International Conference on Machine Learning, 2017, pp. 2208–2217.

[17] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, A. Smola, A kernel two-sample test, J. Mach. Learn. Res. 13 (2012) 723–773.

[18] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, arXiv preprint arXiv:1701.07875 (2017). .

[19] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. C. Courville, Improved training of wasserstein gans, in: Advances in Neural Information Processing Systems, 2017, pp. 5767–5777. .

[20] J. Shen, Y. Qu, W. Zhang, Y. Yu, Wasserstein distance guided representation learning for domain adaptation, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[21] D. Das, C.G. Lee, Sample-to-sample correspondence for unsupervised domain adaptation, Eng. Appl. Artif. Intell. 73 (2018) 80–91.

[22] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, L. D. Jackel, Handwritten digit recognition with a back-propagation network, in: Advances in Neural Information Processing Systems, 1990, pp. 396–404. .

[23] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551.

[24] D.H. Hubel, T.N. Wiesel, Receptive fields of single neurones in the cat's striate cortex, J. Physiol. 148 (3) (1959) 574–591.

[25] A.M. Saxe, P.W. Koh, Z. Chen, M. Bhand, B. Suresh, A.Y. Ng, On random weights and unsupervised feature learning, in: Proceedings of the 28th International Conference on International Conference on Machine Learning, Omnipress, 2011, pp. 1089–1096.

[26] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436.

[27] T.P. Vogl, J. Mangis, A. Rigler, W. Zink, D. Alkon, Accelerating the convergence of the back-propagation method, Biol. Cybern. 59 (4–5) (1988) 257–263.

[28] C. Cheng, G. Ma, Y. Zhang, M. Sun, F. Teng, H. Ding, Y. Yuan, Online bearing remaining useful life prediction based on a novel degradation indicator and convolutional neural networks (2018). .

[29] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448. .

[30] C. Villani, Optimal Transport: Old and New, vol. 338, Springer Science & Business Media, 2008.

[31] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. 17 (1) (2016) 2096, 2030.

[32] D. Sejdinovic, B. Sriperumbudur, A. Gretton, K. Fukumizu, Equivalence of distance-based and rkhs-based statistics in hypothesis testing, Ann. Stat. (2013) 2263–2291.

[33] L. Guo, Y. Lei, S. Xing, T. Yan, N. Li, Deep convolutional transfer learning network: a new method for intelligent fault diagnosis of machines with unlabeled data, IEEE Trans. Industr. Electron. (2018).

[34] S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, IEEE Trans. Neural Networks 22 (2) (2011) 199–210.

[35] M. Long, J. Wang, G. Ding, J. Sun, P.S. Yu, Transfer feature learning with joint distribution adaptation, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2200–2207.

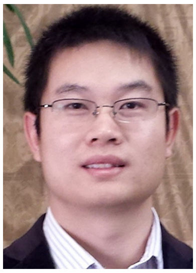
[36] B. Sun, J. Feng, K. Saenko, Return of frustratingly easy domain adaptation, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[37] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: a system for large-scale machine learning, in: Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation, USENIX Association, 2016, pp. 265–283.

Cheng Cheng received the B. Eng. in Measurement, Control Technology and Instrument in 2012 from Tianjin University, China. In 2013 and 2018, she respectively received the MSc and the Ph.D. in Control Systems from Imperial College London, UK. Since 2018, she has been a Postdoctoral Researcher at Huazhong University of Science and Technology, China. Her research interests include robust control, mechatronic systems modelling and simulation, and deep learning applications.

Beitong Zhou received the B. Eng. in Electrical Engineering and Automation in 2017 from Chongqing University, China. Since 2018, he has been a graduate Student at Huazhong University of Science and Technology, China. His research interests include statistical learning, deep learning applications and optimizations.

Guijun Ma received his Bachelor's degree from Huazhong University of Science and Technology, Wuhan, China, in 2017. He is currently working toward the Ph.D. degree in mechanical engineering in School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China. His research interests include machine fault diagnosis and remaining useful life prediction.



Dongrui Wu received the PhD degree in electrical engineering from the University of Southern California in 2009. He was a Lead Researcher at GE Global Research, and Chief Scientist of several startups. He is now a professor in the School of Automation, Huazhong University of Science and Technology, Wuhan, China. His research interests include affective computing, brain-computer interface, computational intelligence, and machine learning. He received the IEEE CIS Outstanding PhD Dissertation Award in 2012, the IEEE Transactions on Fuzzy Systems outstanding paper award in 2014. He is an associate editor of IEEE Transactions on Fuzzy Systems, IEEE Transactions on Human-Machine Systems, and IEEE Computational Intelligence Magazine.

Ye Yuan received the B.Eng. degree (Valedictorian) from the Department of Automation, Shanghai Jiao Tong University, Shanghai, China, in 2008, and the M.Phil. and Ph.D. degrees from the Department of Engineering, University of Cambridge, Cambridge, U.K., in 2009 and 2012, respectively. He has been a Full Professor at the Huazhong University of Science and Technology, Wuhan, China since 2016. Prior to this, he was a Postdoctoral Researcher at UC Berkeley, a Junior Research Fellow at Darwin College, University of Cambridge. His research interests include system identification and control with applications to cyber-physical systems. Dr. Yuan has received the China National Recruitment Program of 1000 Talented Young Scholars, the Dorothy Hodgkin Postgraduate Awards, Microsoft Research Ph.D. Scholarship, Best of the Best Paper Award at the IEEE Power and Energy Society General Meeting, Best Paper Finalist at the IEEE International Conference on Information and Automation.