WDA
An Improved Wasserstein Distance-Based Transfer Learning Fault Diagnosis Method

Zhu, Zhiyu; Wang, Lanzhi; Peng, Gaoliang; Li, Sijue

Published in:
Sensors

Published: 01/07/2021

Document Version:
Final Published version, also known as Publisher's PDF, Publisher's Final version or Version of Record

Published version (DOI):
10.3390/s21134394

Publication details:
Zhu, Z., Wang, L., Peng, G., & Li, S. (2021). WDA: An Improved Wasserstein Distance-Based Transfer Learning Fault Diagnosis Method. *Sensors*, *21*(13), [4394]. https://doi.org/10.3390/s21134394

sensors

MDPI

Article

WDA: An Improved Wasserstein Distance-Based Transfer Learning Fault Diagnosis Method

Zhiyu Zhu [1,2], Lanzhi Wang [3], Gaoliang Peng [1,*] and Sijue Li [1]

[1] State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China; zhiyuzhu2-c@my.cityu.edu.hk (Z.Z.); lisijue@hit.edu.cn (S.L.)
[2] The Department of Computer Science, City University of Hong Kong, Hong Kong
[3] Beijing Institute of Aerospace Launch Technology, Beijing 100076, China; bluewill-926@163.com
* Correspondence: pgl7782@hit.edu.cn

Abstract: With the growth of computing power, deep learning methods have recently been widely used in machine fault diagnosis. In order to realize highly efficient diagnosis accuracy, people need to know the detailed health condition of collected signals from equipment. However, in the actual situation, it is costly and time-consuming to close down machines and inspect components. This seriously impedes the practical application of data-driven diagnosis. In comparison, the full-labeled machine signals from test rigs or online datasets can be achieved easily, which is helpful for the diagnosis of real equipment. Thus, we introduced an improved Wasserstein distance-based transfer learning method (WDA), which learns transferable features between labeled and unlabeled signals from different forms of equipment. In WDA, Wasserstein distance with cosine similarity is applied to narrow the gap between signals collected from different machines. Meanwhile, we use the Kuhn–Munkres algorithm to calculate the Wasserstein distance. In order to further verify the proposed method, we developed a set of case studies, including two different mechanical parts, five transfer scenarios, and eight transfer learning fault diagnosis experiments. WDA reached an average accuracy of 93.72% in bearing fault diagnosis and 84.84% in ball screw fault diagnosis, which greatly surpasses state-of-the-art transfer learning fault diagnosis methods. In addition, comprehensive analysis and feature visualization are also presented.

Keywords: intelligent bearing fault diagnosis; Wasserstein distance; convolutional neural network; domain adaptive ability; Kuhn–Munkres algorithm

Citation: Zhu, Z.; Wang, L.; Peng, G.; Li, S. WDA: An Improved Wasserstein Distance-Based Transfer Learning Fault Diagnosis Method. *Sensors* **2021**, *21*, 4394. https://doi.org/10.3390/s21134394

Academic Editors: Kim Phuc Tran, Athanasios Rakitzis and Khanh T. P. Nguyen

Received: 12 June 2021
Accepted: 23 June 2021
Published: 26 June 2021

1. Introduction

With the rise of machine learning, especially deep learning, more and more data-driven algorithms have been proposed and applied successfully in different fields in the last few years [1–3]. Similarly, data-driven methods are increasingly suggested to deal with problems in the field of machine health monitoring [4], which has great importance in modern industry.

For example, Atoui et al. [5] presented Bayesian network for fault detection and diagnosis, Rajakarunakaran S et al. [6] proposed artificial neural networks (ANN) for the fault detection of the centrifugal pumping system, and Ivan et al. [7] suggested a novel weighted adaptive recursive fault diagnosis method based on principal component analysis (PCA) to reduce the false alarm rate in processing monitoring schemes. Recently, as deep learning is rapidly developing, artificial intelligence methods are considered to handle the fault detection and classification in rolling bearing elements, e.g., autoencoders [8] and convolutional neural networks (CNN). Li et al. [9] proposed a bearing defect diagnosis technique based on a fully connected winner-take-all autoencoder. Jafar Zarei [10] proposed a pattern recognition technique for fault diagnosis of induction motor bearings via utilizing the artificial multilayer perceptron neural networks. Olivier Janssens et al. [11] introduced feature learning means for condition monitoring based on convolutional neural networks

to obtain signal features for bearing fault detection. Although many studies have been conducted, most of them are only effective under a large amount of labeled data.

Through a brief review, it is obvious that most methods are only confirmed in theory, and few are able to be applied in industry [12,13]. In a real industry situation, as machines usually work in a healthy state, it is quite a difficult task to determine whether a fault has occurred during the data collection. Moreover, even when the equipment breaking down is known, it is difficult to point out the definite fault types disassembling and inspecting the components such as bearings and ball screws in a machine, being time and labor-consuming tasks. Additionally, the real machine always works under various working conditions. Thus, the collected signals are combined with different data distributions. Those scenarios in the industrial applications will seriously impact the performance of data- based fault diagnosis.

To overcome problems of different working conditions, some researchers proposed increasing the generalization ability of the algorithms, which are named domain adaption techniques. For instance, Zhang et al. [14] proposed a deep neural network with high diagnostic accuracy in diagnosing signals with high noise and signals from different loads. In their work, the authors suggested that the high-level features of data from the different working conditions have a more similar distribution and are less affected by noise. Moreover, Zhu et al. [15] used capsule net to extract more general features from the time-frequency spectrum and achieved higher diagnosis accuracy when dealing with data from different loads. With such improvement strategies, artificial neural networks have been proven to be a potential tool to deal with industry data. However, the above methods only focus on the variation between working conditions (e.g., speed, loads) on one machine, and they cannot handle the huge variations of mechanism between different types of equipment.

Transfer learning theory has been introduced to machine fault diagnosis in order to improve domain adaption ability among different machines. Transfer learning aims to reduce the distribution discrepancy of diverse domains, as data from the target domain have similar knowledge but different distribution compared to the source domain. For example, Lu et al. [16] presented a deep model-based domain adaptation method for the machine fault diagnosis. A gearbox dataset collected under different operation conditions was used to test the performance of the proposed method. Wen et al. [17] set up a new deep transfer learning method for fault diagnosis. The validation dataset was acquired from a bearing testbed operating under different working conditions. Xie et al. [18] proposed a transfer analysis-based gearbox fault diagnosis method. The performance of the presented method was verified by a gearbox dataset obtained under various operation conditions. Guo et al. [19] proposed deep transfer learning-based methods using maximum mean discrepancy and adversarial training techniques together to regularize the discrepancy between different domains. Sandeep et al. [20] presented a ConvNet-based transfer learning method for bearing fault diagnosis with varying speeds. Hasan et al. [21] proposed a transfer learning fault diagnosis framework using 2D acoustic spectral imaging-based pattern formation method. Zhang et al. [22] introduced hybrid-weighted adversarial learning to address the domain adaptation problem. Meanwhile, Zhang et al. [23] also utilized federated learning to facilitate the mechanical fault diagnosis. However, the above transfer learning methods took advantage of enough labeled data. Unfortunately, labeled signals from the practical industrial machine are rare and hard to collect.

As the most critical issue during the process of transfer learning, modeling and optimizing the discrepancy between different domains are the core of the proposed method. As a stable and continuous measurement, Wasserstein distance has displayed its superiority in different applications, e.g., image generation [24,25]. Thus, in this paper, we propose a new method with excellent domain adaptive ability based on Wasserstein distance (WDA) in order to deal with machine fault data from different machines. Cosine similarity and the Kuhn–Munkres algorithm are introduced to improve transfer effects. The contributions of this paper mainly lie in the following two parts:

(1) To achieve classification on unlabeled signals, we propose a transfer learning fault diagnosis method named WDA, which makes use of labeled signals from different machines to help the classification of signals. In WDA, Wasserstein distance is applied to manage the gaps between two distributions, during which we utilize cosine similarity to measure the discrepancy between feature embeddings. Moreover, Kuhn-Munkres algorithm is introduced to directly optimize the Wasserstein distance.

(2) We carried out extensive experiments to validate the effectiveness of the proposed method on various transfer scenarios. Meanwhile, to better illustrate the training process of high-dimensional feature embeddings, we also visualized the whole training process.

The structure of this paper is organized as follows. In Section 2, we introduce the basic conception of transfer learning, Wasserstein distance, and the corresponding Kuhn-Munkres algorithm. Following that, the proposed method and optimization algorithm are discussed in Section 3. Then, the experiments are carried out in Section 4 to verify the proposed method. Finally, the conclusion is drawn from the above experiments.

2. Related Works

In the field of machine learning, transfer learning is proposed to deal with the differences between the signals from the source domain and target domain, while Wasserstein distance is a powerful criterion of the discrepancy. However, the calculation of Wasserstein distance belongs to the general assignment problem. Yet, in most of the research work [26–28], there has hardly been one direct calculation of it. Thus, a brief introduction of transfer learning, Wasserstein distance, and the solution of the general assignment problem (GAP) are helpful to know about the development and the limitation of recent works.

2.1. Transfer Learning

Transfer learning is different from many other traditional machine learning methods, which are established under the assumption that training data and test data are drawn from the same distributions. To better illustrate transfer learning, we introduce two important conceptions: domain and task, as follows [29].

To begin with, domain D includes two key components: feature space χ and marginal distribution $P(X)$, where $X = \{x_1, \ldots, x_n\} \in \chi$ means that X is a set containing samples from feature space χ, e.g., the signals collected from the machine in different health conditions. Then, a task consists of two components: a label space Y and an objective function $G(\cdot)$, corresponding to the health conditions of signals and classification algorithm. Generally speaking, the objective function could not be directly observed. However, it could be learned from training data, which consist of pairs $\{x_i, y_i\}$. with the notion of source domain data $D_s = \{(x_{S1}, y_{S1}), \ldots, (x_{Sns}, y_{Sns})\}$ and target domain data $D_T = \{(x_{T1}, y_{T1}), \ldots, (x_{Tnt}, y_{Tnt})\}$. The transfer learning could be defined as the following:

Given source domain D_S and learning task T_S, a target domain D_T and learning task T_T transfer learning aims to help improve the performance of the predictive function $f_T(\cdot)$ in D_T through using the knowledge in D_S and T_S, where $D_S \neq D_T$ or $T_S \neq T_T$.

In the field of fault diagnosis, source and target domains usually are different. However, the tasks are equivalent, i.e., $D_S \neq D_T$, $T_S = T_T$. This kind of problem is also called domain adaptation, belonging to transductive transfer learning [29,30]. For the transfer learning problems, there are four different approaches to solve them: instance transfer, feature representation transfer, parameter transfer, and relational knowledge transfer. Among them, the feature representation transfer is a widely used transfer learning method in transfer fault diagnosis [18,19,31–33]. Moreover, there are currently two methods to bridge the gap between two distributions: feature extractor regularization, applying regularization terms on feature extractor to obtain features extracted from different domains in similar distributions, or using adversarial training methods to close two distributions.

Firstly, maximum mean discrepancy [34,35] and Wasserstein distance are widely used to measure discrepancies in domain adaptation transfer learning. They are used to regularize the output feature of the feature extractor to obtain equivalent marginal

distribution. Secondly, some adversarial training methods such as DANN [36] are also proposed to narrow the gap between source and target domain. Most of them use adversarial training techniques in artificial neural networks to manage the gap of two different distributions. However, these training methods suffer problems, e.g., those methods are hard to train [37,38] and converge to a high-performance result. Thus, a high accuracy method is badly needed.

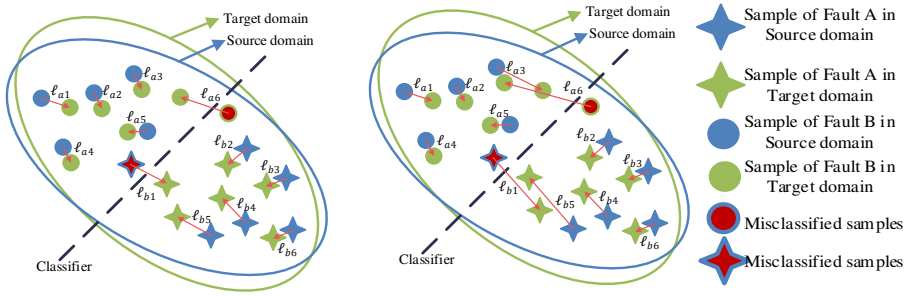
2.2. Wasserstein Distance

Wasserstein distance, also called earth mover's distance, is a metric to measure the discrepancy between two distributions, and it is widely used in domain adaptation, e.g., WGAN [24] and BEGAN [39]. Wasserstein distance is generally based on a way that transforms one distribution to the other with minimal cost.

As shown in Figure 1a, different discrepancies of two domains are represented, which could also be considered as the cost of transporting distribution from one domain to the other. We define the transporting cost as:

$$\ell = \frac{1}{n_a + n_b}\left(\sum_{i=1}^{n_a}\mathcal{F}(\ell_{ai}) + \sum_{j=1}^{n_b}\mathcal{F}\left(\ell_{bj}\right)\right) \tag{1}$$

where n_a, n_b denote the numbers of samples of different fault types, and $\mathcal{F}(\cdot)$ represents a function measuring the difference between two samples, which usually is L2-norm or L1-norm. As shown in Figure 1b, Wasserstein distance (noted as ℓ_1) is used to transport the feature from the source domain to the target domain with minimal cost. The other transport method, e.g., ℓ_2, shown in Figure 1b, is higher than ℓ_1.



(**a**) Pairs of Wasserstein distance (**b**) Pairs of a high-cost transfer method

Figure 1. Different pairs for measuring discrepancy between two distributions.

The formula of Wasserstein distance (ℓ_w) is shown as:

$$\ell_w = inf_{\gamma\in\Gamma(\mu_1,\mu_2)}E_{(x_1,x_2)\sim\gamma}\ell(x_1,x_2) \tag{2}$$

From the above equation, we can see that the Wasserstein distance is a low bound of the cost to transform a distance between two distributions. Berthelot et al. also proposed BEGAN to optimize the lower bound of Wasserstein distance to achieve better performance on image generation [39]. Note that all the above methods are unsupervised methods. Different from supervised or semi-supervised methods, unsupervised methods do not care about the similarity of distributions of the input and output domains. However, it would remain a huge problem, especially in the beginning training stage, if the discriminator is extremely unstable. Moreover, it is difficult to use it to regularize the feature extractor. Moreover, the discriminator could only be said to safely match the 1-Lipschitz function in the features that already are trained with the discriminator. That is, with the training process going on, the distribution of the high-level features may change, and that discriminator may not correctly calculate the distance between features from two domains. Thus, the methods based on adversarial training struggle to achieve high performance.

2.3. General Assignment Problem and Kuhn–Munkres Algorithm

The calculation of Wasserstein distance belongs to a general assignment problem (GAP) while the samples of two distributions are equal. Considering that there are m samples from source domain $\{x_{S1}, \ldots, x_{Sm}\}$ and n samples $\{x_{T1}, \ldots, x_{Tn}\}$ from target domain, without loss of generality, we assume that $n \leq m$. Any target samples x_{Tj} could be assigned to the source x_{Si}. Each pair $\{x_{Si}, x_{Tj}\}$ has a cost $c(x_{Si}, x_{Tj})$ to transfer from x_{Tj} to x_{Si}. The task is to assign n target samples to m source samples with the minimal cost, which is also the Wasserstein distance between two distributions. Moreover, the assignment problem could be formulated as the following optimization problem:

$$\min \sum_{i=1}^{\Updownarrow} \sum_{j=1}^{\setminus} \rfloor(x_{Si}, x_{Tj}) \cdot \mathcal{T}_{i,j} s.t.\ 0 \leq \sum_{i=1}^{m} \mathcal{T}_{i,j} \leq 1,\ \sum_{j=1}^{\setminus} \mathcal{T}_{i,j} = 1,\ \mathcal{T}_{i,j} \in \{0,1\}. \tag{3}$$

The K-M algorithm [40] could be implanted through different versions: graph [41,42] or matrix [43]. Unlike the adversarial learning-based methods, which utilize discriminator to approximate Wasserstein distance of two distributions [26,28], in this section, we introduce the K-M algorithm through graph perspective, which has been applied to the applications such as multi-objective optimization [44] and role transfer [45]. Considering a bipartite graph $G = (X_{Si}, E, X_{Tj})$, where E means the edges of pairs (x_{Si}, x_{Tj}) and $E \in X_{Si} \times X_{Tj}$, we introduce the following three definitions:

Definition 1: *Neighborhood: the neighborhood of a vertices x is the set $\mathcal{I}_G(x)$ with all vertices sharing edges with x; similarly, the neighborhood of a set X is $\mathcal{I}_G(X)$, whose all vertices are sharing edges with any vertices in X.*

Definition 2: *Feasible label: it is a function $\ell : X \to R$, which satisfies the following condition:*

$$\ell(x_{Si}) + \ell(x_{Tj}) \geq w(x_{Si}, x_{Tj})\ \forall x_{Si} \in X_S\ \forall x_{Ti} \in X_T \tag{4}$$

Definition 3: *Matched/exposed: considering a match M, the vertex x is called matched if it is a vertex in M. Otherwise, it is exposed.*

Meanwhile, G_l denotes the subgraph of G, which contains those edges that perfectly satisfy the feasible label, such as the following:

$$\ell(x_{Si}) + \ell(x_{Tj}) = w(x_{Si}, x_{Tj}) \tag{5}$$

Moreover, G_l contains all the vertices of G. The K-M Algorithm 1 for solving the assignment problem is shown below.

The K-M algorithm can efficiently address assignment problems, especially small-scale ones, e.g., transfer between two mini-batch samples. Meanwhile, Wasserstein distance as a useful divergence to measure the distance between two distributions has been widely used in the field of transfer learning. However, the performances of these methods leave much to be desired. Most of them used the approximation form of Wasserstein distance instead of calculating it directly. Actually, the calculation of Wasserstein distance is an assignment problem that could compute through the K-M algorithm. Thus, we proposed a novel method using the K-M algorithm to address the discrepancy measurement of transferring between two domains.

Algorithm 1. Kuhn–Munkres Algorithm

Input: A bipartite graph G = (X_S, E, X_T), corresponding edge weights $\exists\left(x_{Si}, x_{Tj}\right)$

Output: the perfect matching M.

Step 1: Generate initial labeling ℓ and match in G_ℓ

Step 2: If M perfect, stop. Otherwise, pick a free vertex $x_{Si} \in X_S$. Set S = x_{Si}, T = $\varnothing$.

Step 3: If $\mathcal{I}_S(X) =$ T, update labels (forcing $\mathcal{I}_S(X) \neq$ T) with following Equations (6) and (7)

$$\alpha_\ell = \min_{s \in S, y \notin T}\left\{\mathscr{O}(x_{Si}) + \mathscr{O}\left(x_{Tj}\right) - w\left(x_{Si}, x_{Tj}\right)\right\} \tag{6}$$

$$\hat{\mathscr{O}} = \begin{cases} \mathscr{O}(\mathrm{x}) - \alpha_\ell, & x \in S \\ \mathscr{O}(\mathrm{x}) + \alpha_\ell, & x \in T \\ \mathscr{O}(\mathrm{x}), & otherwise \end{cases} \tag{7}$$

Step 4: If $\mathcal{I}_S(X) \neq$ T, choose y $\in \mathcal{I}_S(X) - T$:

If y free, $u - y$ is augmenting path. Augment M and go to 2

If y matched, say to x, extend alternating tree: S = S $\cup$ x, T = T $\cup$ y. Go to

3. Proposed Method

In this section, the proposed Wasserstein distance-based domain adaptive neural network (WDA) is discussed. The architecture of the neural network and the objective of WDA are introduced.

The framework of the proposed method is shown in Figure 2. Meanwhile, the detailed architecture is shown in Figure 3. WDA is composed of two parts: CNN (feature extractor) and a fully connected layer to extract features (noted as $G_1\left(\theta_f, \cdot\right)$), and a full-connected layer (classifier) noted as $G_2(\theta_c, \cdot)$. The aim of CNN is to extract high-level features from input data. Before high-level features are fed into the classifier, Wasserstein distance is used to regularize the features from two different domains. Thus, the CNN could extract features from different domains with similar distributions. Finally, the classifier is used to predict the health conditions of different signals.

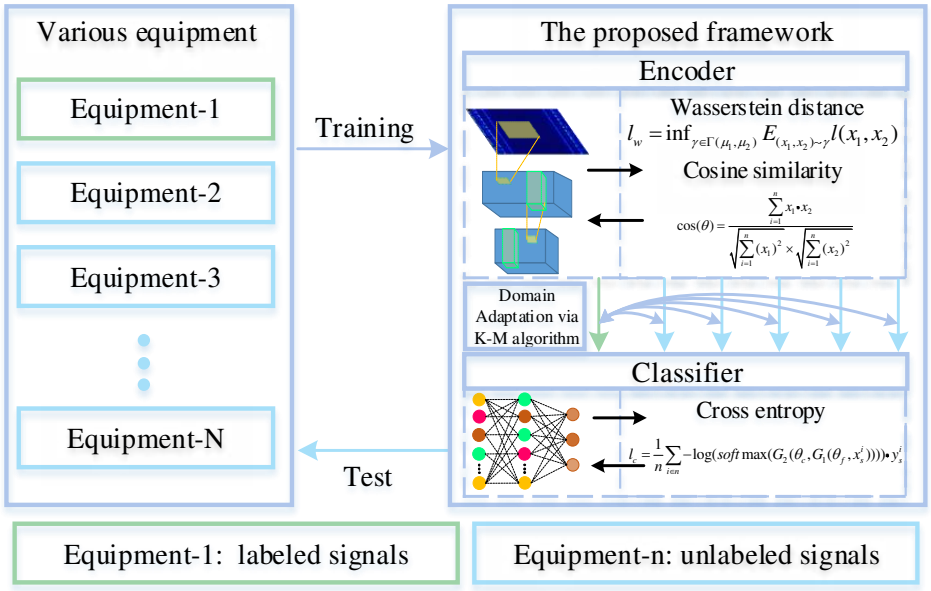


Figure 2. Flow chart of the proposed framework for fault diagnosis on different working equipment. During the training phase, we narrow the gaps between the distributions. At the test phase, the WDA directly predicts the health conditions of unlabeled signals.

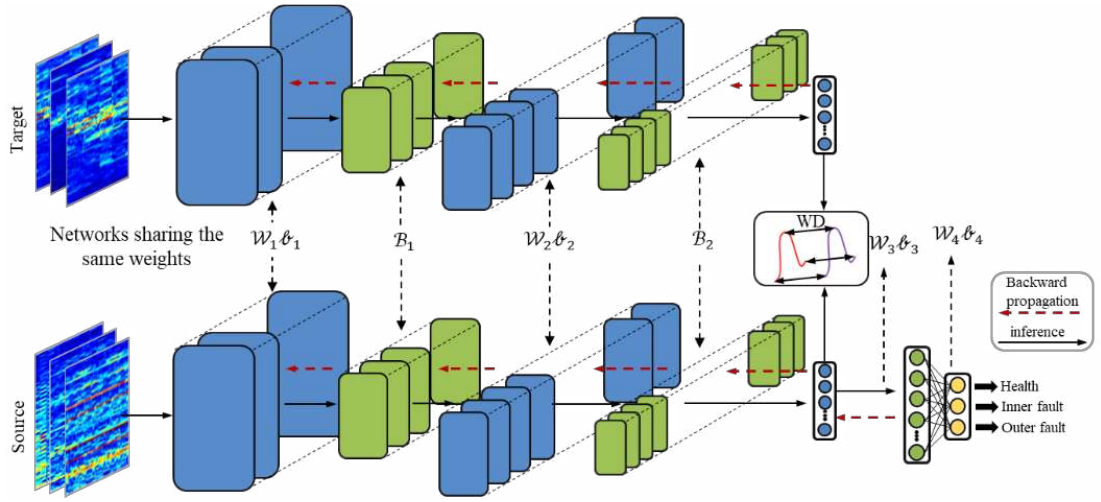


Figure 3. The architecture of the proposed WDA. In the testing phase, the feature maps from the target domain are directly fed into the classifier.

3.1. Network Architecture

The architecture of WDA is shown in Figure 4. It contains two parts: feature extractor and fully connected classifier.

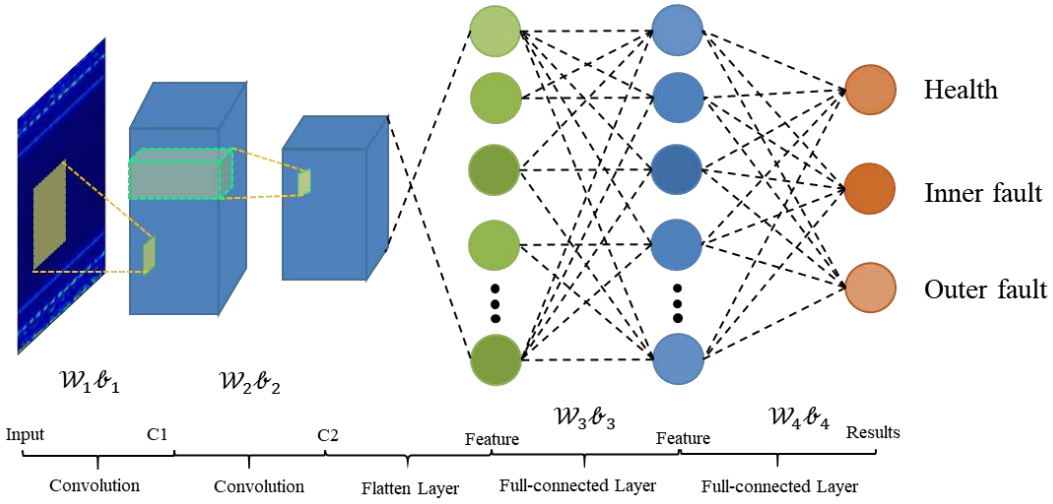


Figure 4. Architecture of WDA.

As shown in Table 1, there are essentially 12 layers in the proposed WDA. The feature extractor block contains two Conv-BN-Pooling-activation modules and a full-connected layer. In addition, the classifier contains only one full-connected layer to predict the health conditions of input data. The details of WDA are shown in Table 2. Due to the different conditions of these methods, the ranges of labels vary from condition to condition, e.g., in some datasets, they are health, inner fault, and outer fault. However, in other datasets, there are four fault types: health, inner fault, outer fault, and rolling ball fault, where $\mathcal{N}_o$ means the number of classes of the models, and it should be 3 or 4.

Table 1. Details of the classifier.

No.	Layer Name	Kernel Size/Stride/Filters	Parameters	Symbols	Output Shape
1	Convolution1	$4 \times 4/1/16$	$16(4 \times 4 \times 1 + 1) = 202$	$\mathcal{W}_1\ell_1/\mathcal{B}_1$	(60,60,16)
2	BatchNorm1	-	$16 \times 2 = 32$		(60,60,16)
3	MaxPooling1	$4 \times 4/1/1$	-		(30,30,16)
4	ReLU	-	-		(30,30,16)
5	Convolution2	$3 \times 3/1/64$	$64 \times 16 \times (3 \times 3 + 1) = 10{,}240$	$\mathcal{W}_2\ell_2/\mathcal{B}_2$	(28,28,64)
6	BatchNorm2	-	-		(28,28,64)
7	MaxPooling2	$2 \times 2/2/1$	-		(14,14,64)
8	ReLU2	-	-		(14,14,64)
9	Dense Layer1	-	$(14 \times 14 \times 64 \times 96 + 1) = 1{,}204{,}225$	$\mathcal{W}_3\ell_3$	96
10	BatchNorm3	-	-		96
11	ReLU	-	-		96
12	Dense Layer2	-	$(128 \times 3\ (4) + 1) = 387/(513)$	$\mathcal{W}_4\lfloor_4$	$\mathcal{N}_o$

Table 2. Description of the dataset.

Dataset	Sample Rate	Resample Rate	Loads	Speed
IMS	20 KHz	1 KHz	6000 lbs.	2000 RPM
Self-collected	25 KHz	1 KHz	0 lbs.	900–1500 RPM
CWRU	48 KHz	1 KHz	2 hp	1750 RPM

3.2. Objective of WDA

In the proposed WDA, the loss function consists of two parts: classification loss (ℓ_c) on the source domain D_S and domain adaptive loss (ℓ_w) between source D_S and target domains D_T. The classification loss aims at reducing the classification error on the source domain, and the domain adaptive loss aims to bridge the gap between the source domain and target domain. In the following section, they are introduced separately.

3.2.1. Classification Loss

Classification loss of WDA is a cross-entropy loss set as Equation (8), where softmax is described in Equation (9). As shown in Algorithm 1, $G_1\left(\theta_f, \cdot\right)$ represents the feature extractor and $G_2(\theta_c, \cdot)$ represents the classifier. Note that classification loss is only acted upon a source domain data whose labels are known.

$$\ell_c = \frac{1}{n}\sum_{i \in n} -\log\left(softmax\left(G_2\left(\theta_c, G_1\left(\theta_f, x_s^i\right)\right)\right)\right) \cdot y_s^i \tag{8}$$

$$softmax(z_{i0}) = \frac{exp(z_{i0})}{\sum_{i \in num\_class} exp(z_i)} \tag{9}$$

3.2.2. Domain Adaptive Loss

Usually, for the semi-supervised problems, most methods want to regularize the feature extractor to obtain the features of source and target domains in exactly the same distribution. However, it is too strict for transfer learning categorical models. Actually, for the classifier, especially the linear classifier, the real concern is the pattern of the features (e.g., orientation of features). We demonstrate it through the following equation:

$$f(W, b, s_i) = Ws_i + b \tag{10}$$

As shown in Equation (10), a linear classifier, which is designed to classify output features from a feature extractor, is present to explain the mechanism, where $W \in R^{n,m}$

indicates the weights of classifier, $b \in R^m$ means the bias of the classifier, and $s_i \in R^n$ indicates the input feature of the classifier. If the label of the feature s_i is y_i, Equation (11) would be established:

$$f(W, b, s_i)_{yi} > f(W, b, s_k)_{yk}\ y_k \neq y_i. \tag{11}$$

As we utilize the ReLu activation function, there is an interesting characteristic that $ReLU(\alpha s_i) = \alpha ReLU(s_i)$ (for $s_i \leq 0$, $ReLU(\alpha s_i) = 0 = \alpha ReLU(s_i)$). From Equation (12), we can see that if the label of the feature s_i is y_i, the prediction of feature $\alpha \cdot s_i$ in the same orientation with s_i is also y_i, that is:

$$f(w, b, \alpha \cdot s_i)_{yi} = \alpha \cdot f(W, b, s_i)_{yi} > \alpha \cdot f(w, b, s_k)_{yk} = f(w, b, \alpha \cdot s_k)_{yk}. \tag{12}$$

Equation (12) shows that the scale of features actually does not affect the classification result. Thus, the traditional ways are limited by using the L2-norm to measure the disparities between two variables. It is noted that cosine similarity calculates the orientation divergence of two vectors, which focuses more on the output pattern. Thus, for the feature extractor, we could use the cosine similarity to measure samples from different domains and change Wasserstein distance as:

$$\ell_w = inf_{\gamma \in \Gamma(\mu_1, \mu_2)} E_{(x_1, x_2) \sim \gamma} \mathbb{C}(x_1, x_2) \tag{13}$$

where $\mathbb{C}(x_1, x_2)$ is cosine similarity from feature x_1, x_2, shown as the following:

$$\mathbb{C}(x_1, x_2)\mathbb{C}(x_1, x_2) = arccos\left(\frac{x_1 \cdot x_2}{\|x_1\| \cdot \|x_2\|}\right)\frac{1}{\pi} \tag{14}$$

Once the objectives and architecture of WDA are established, the optimization of the proposed method is introduced in the following section.

3.3. Optimization of WDA

Following the establishment of the architecture and objective of WDA, the training algorithms are introduced in this chapter. The optimization algorithm is shown in Algorithm 2.

Moreover, in order to verify the choice of cosine similarity, we carried out experiments that contain feature visualization and comparisons with state-of-the-art domain adaptive transfer learning methods.

Algorithm 2. Training WDA with ADAM optimization method N_c = number of categories

Initialize: initial WDA feature extractor parameters θ_f and classifier parameters θ_g
For the number of training iterations, do:
• Sample minibatch of samples $\left(X_S^i, Y_S^i\right) = \left(\{x_S^1 \ldots x_S^n\}, \{y_S^1 \ldots y_S^n\}\right)$, from source domain signals distribution $P_S(X, Y)$, $X_T^j = \{x_T^1 \ldots x_T^n\}$ from target domain $P_T(X)$.
• Extract feature from two different domains with two shared weights feature extractors through Equation (15).

$$\begin{cases} \chi_S^i = G_1\left(\theta_f, X_S^i\right) \\ \chi_T^i = G_1\left(\theta_f, X_T^i\right) \end{cases} \tag{15}$$

• Calculate the $n \times n$ cost matrix A between the high-level features from source and target domains (Equation (16)).

$$A[i, j] \leftarrow \mathbb{C}\left(\chi_S^i, \chi_T^j\right) \; for \; i\epsilon[1, N], j\epsilon[1, N] \tag{16}$$

• Use the K-M algorithm in Table 2 to address the assignment problem of cost matrix A.
Input: bipartite graph $G = \left(X_S, E, X_T = \chi_T^i\right)$,

$$\begin{cases} X_S = \{\chi_S^1, \chi_S^2, \ldots, \chi_S^n\} \\ X_T = \{\chi_T^1, \chi_T^2, \ldots, \chi_T^n\} \\ E = A \end{cases} \tag{17}$$

Output: permutations S.
After obtaining optimal permutations $S = \{S_1, S_2 \ldots S_n\}$, calculate Wasserstein distance ℓ_d:

$$\ell_d \leftarrow \sum_{i=1}^{n} A[i, S_i] \tag{18}$$

• Calculate cross-entropy classification loss on the source domain.

$$\ell_c \leftarrow \frac{1}{n} \sum_{i=1}^{n} -\log\left(softmax\left(G_2\left(G_1\left(x_S^i, \theta_f\right), \theta_g\right)\right)\left[y_S^i\right]\right) \tag{19}$$

• Calculate cross-entropy loss on the source domain.
• Calculate loss.

$$\ell = \ell_c + \ell_d \tag{20}$$

• Backward propagation of ℓ, getting the gradients of parameters and updating the parameters θ_f, θ_g.
end

4. Case Study and Experiment Result

In this section, experiments and analyses of the model that were carried out are shown. In order to verify the generalization of the proposed method, we separately investigated transfer scenarios on different mechanical parts, bearing and ball screws. WDA was written in python 3.6, Pytorch 0.4.1 training with Intel i3-8100 CPU, and a GTX1070 GPU.

4.1. CASE I: Bearing Fault Diagnosis

In this section, the proposed method was trained and tested on three different domains. There were three datasets named IMS dataset (α), self-collected bearing dataset (β), and CWRU bearing dataset (γ). We first give a brief introduction to those three datasets. Then, we present the data preprocessing procedures with implementation details and finally discuss the experimental results.

4.1.1. α: IMS Bearing Dataset

The data were generated by the NSF I/UCR Center for Intelligent Maintenance Systems (IMS) [46]. These sets of data contain four bearings that were run to failure under a constant load as shown in Figure 5a,b. Every 10 min, 1 s vibration signals were collected and saved into a file that contains 20,480 points for each bearing. IMS contains four different conditions: health, inner fault, rolling elements fault, and outer fault. Radial load is 6000 lbs., and rotation speed is kept constant at 2000 RPM under all conditions.

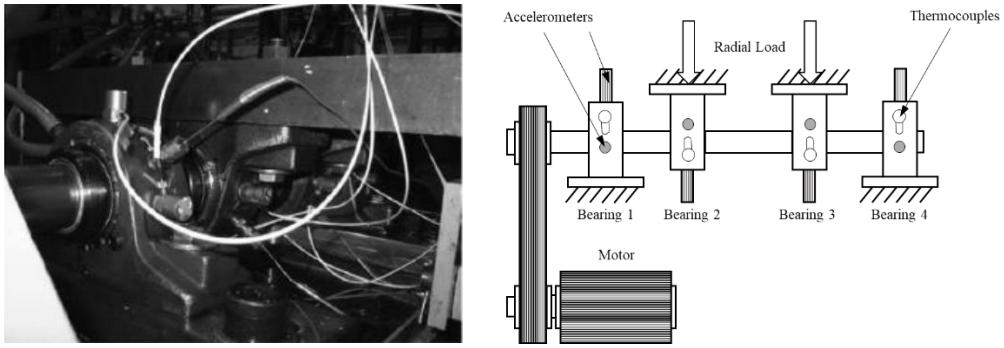


Figure 5. (**a**) Test rig of IMS dataset. (**b**) Illustration of IMS test rig.

4.1.2. β: Self-Collected Bearing Dataset

The second dataset was collected by the test rig shown in Figure 6. It contains an induction motor, an accelerometer, and a rotation shaft with two bearings for support. Bearing is in the type of 6204. The dataset contains three different health conditions: health, inner fault, and outer fault as shown in Figure 7. The dataset includes artificial defects, which are shown in Figure 6. Different rotation speeds were also collected, including 900 RPM, 1020 RPM, 1140 RPM, 1260 RPM, 1380 RPM, and 1500 RPM, while the sample rate was 48 kHz.

Figure 6. Test rig used to collect different speeds and different sample rate data.

(**a**) EMD (**b**) Drilling (**c**)Pitting

Figure 7. Different fault in testing bearings.

4.1.3. γ: CWRU Bearing Dataset

Data from dataset γ were collected from Case Western Reserve University [47], whose test rig is shown in Figure 8. All faults in the dataset arise in the form of EMD. The experimental setup mainly contained an induction motor, an accelerometer, testing bearings, and a loading motor.

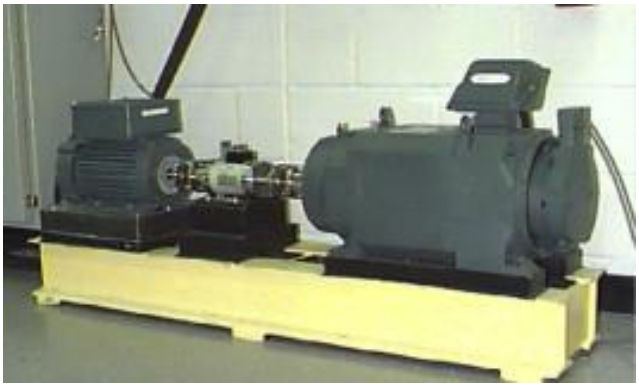


Figure 8. Test rig used in Case Western Reserve University Lab [41].

Each bearing was tested under four different loads (1, 2, and 3 hp). In addition, damages caused by EMD lie in the outer race, inner race, or rollers of the bearings with fault diameters of 0.007, 0.014, and 0.021 inches (1 in. = 25.4 mm), respectively, which means that the number of categories under each load is 10. All of the information is listed in Table 3.

Table 3. The working condition of different loads.

Index	Supporting	Speed (RPM)	Loads (N.M)
ζ	Fixed—floating	$1500 \times \mathrm{Sin}(5t)/400/1500$	0/10/35
η	Fixed—none	$1500 \times \mathrm{Sin}(5t)/400/1500$	0/10/35

Data preprocessing and implementation details: In the proposed method, the short-time Fourier transform is applied to the raw signals to obtain a time-frequency graph. For a window sliding on the raw signals at the same stride, we obtained the signals in the window and applied Fourier transform to it. With the above steps occurring, we could change a series of time-domain signals to a graph that fuses both time and frequency features. In order to reduce the accidental noise, we applied normalization to the time-frequency graph as:

$$x^* = \frac{x - \mu}{\sigma} \tag{21}$$

where x is input signals, and μ and σ are the average and standard deviation of the data, respectively. Through zero-mean normalization, the effect of the noise and zero drift on the data could be removed.

In Figure 9 and Table 2, different datasets contain different signals collected in different sample rates. The sample rate greatly affects the characteristics of signals. Moreover, it is fixed for one dataset and artificially set. Thus, in the experiment, the signals were resampled to be the same (1 Kh). Meanwhile, short time Fourier transformation (STFT) was used as means of preprocessing. The kernel size of STFT was set to 128, and the stride was 5. Moreover, the size of the output time-frequency graph (TFG) was 128×63. Then it was clipped to 63×63 because TFG is symmetrical, and the first element was the dc component. Thus, the length of raw signals of a TFG was $128 + 62 \times 5 = 438$. Due to the fault types of different datasets: β, γ contained four health conditions, α contained three health conditions, and the number of samples in datasets was changed. The sample number of each health condition was 5000, e.g., in the transfer condition $\gamma \rightarrow \alpha$, the numbers of samples in train and test domains both were 20,000; however, in $\alpha \rightarrow \beta$, they were 15,000. All the training and testing signals were randomly sampled from datasets.

During the training phase, we utilized the ADAM optimizer with the first and the second momentums as 0.9 and 0.999, respectively. We trained it for 100 epochs with the batch size set to 128. The learning rate was initialized to 0.003 and exponentially decayed with a factor of 0.98 for each epoch to 0.00013. The comparison methods are listed in Table 4. The detailed information of training process of different methods is presented in Table 5. CNN denotes a simple convolutional network without any transfer learning technique. SVM represents a support vector machine [48], which is also only trained on the source domain. DDC [34] and DANN [36] are image-based transfer learning algorithms. For fair comparison, we trained them with the time-frequency graph (TFG), which is the same as the proposed method. DCTLN [17] is a transfer learning-based deep neural network for bearing fault diagnosis. We trained those methods with the same protocols and recommended hyper-parameters from the original paper for a fair comparison.

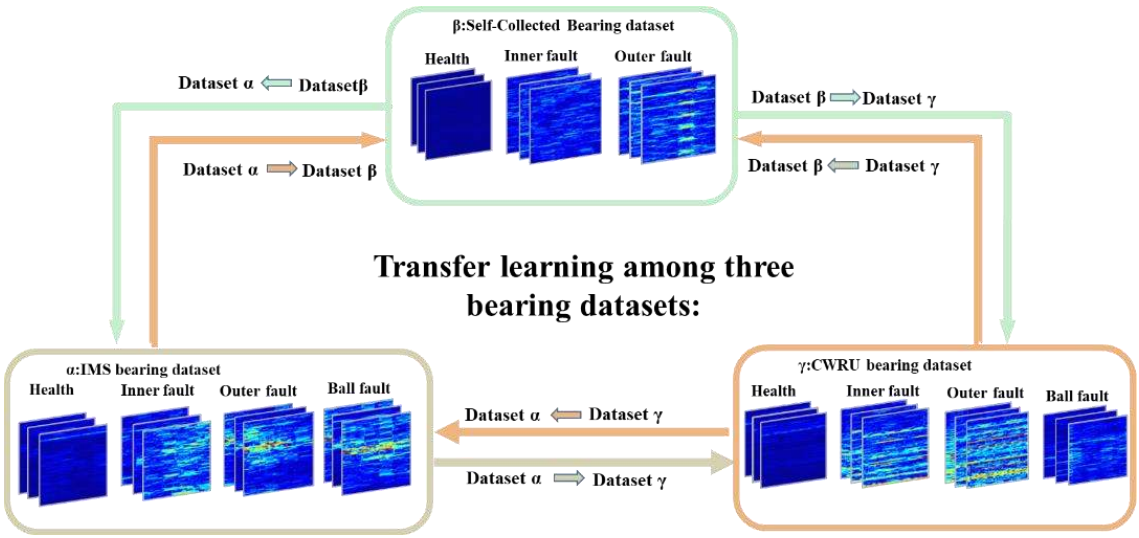


Figure 9. Transfer pipeline of the proposed framework. Dataset $\alpha \rightarrow$ dataset β denotes that we utilized the α as the source domain and the β as the target domain.

Table 4. Accuracies of methods under different transfer conditions.

Method	CNN	SVM	DDC	DANN	DCTLN	WDA
$\zeta \rightarrow \eta$	53.30%	48.78%	67.50%	58.10%	74.34%	**89.13%**
$\eta \rightarrow \zeta$	47.41%	49.60%	63.20%	66.53%	73.25%	**80.54%**
Average	50.36%	49.19%	65.35%	63.32%	73.80%	**84.84%**

Table 5. Description of the training process of different methods.

Name	Property	Input Type
CNN	Supervised (only source domain)	TFG
SVM	Supervised (only source domain)	TFG
DCC	Transfer learning	TFG
DANN	Transfer learning	TFG
DCTLN	Transfer learning	Time frequency signals
WDA	Transfer learning	TFG

The experiment results in Figure 10 and Table 6 show the excellent performance of the proposed method. No matter the traditional machine learning method or deep learning method, it is easy to obtain semi-supervised methods that could achieve better performance than supervised methods. DCTLN as a transfer learning method designed for fault diagnosis showed its superiority over general transfer learning algorithms such as DCC and DANN. However, proposed WDA exceeded DCTLN in most conditions, especially conditions $\gamma \rightarrow \beta$ (from 80.60% to 98.96%). Although DCTLN achieved 89.70% on the $\gamma \rightarrow \alpha$, it only exceeded 1.67% to WDA. Moreover, WDA finally achieved an average accuracy of 93.72%, more than 7.79% of DCTLN, and 16.45% and 22.87% of DANN and DCC, respectively.

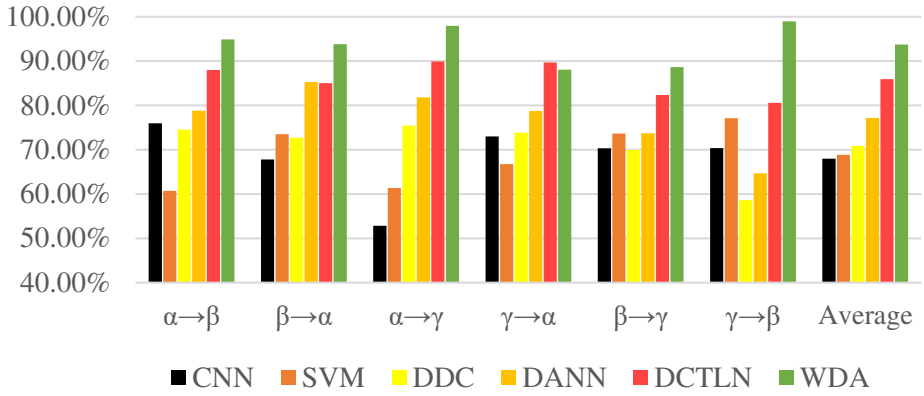


Figure 10. Comparison accuracies of different methods.

Table 6. Results of different transfer result.

Method	$\alpha \rightarrow \beta$	$\beta \rightarrow \alpha$	$\alpha \rightarrow \gamma$	$\gamma \rightarrow \alpha$	$\beta \rightarrow \gamma$	$\gamma \rightarrow \beta$	Average
CNN	75.95%	67.78%	52.83%	73.01%	70.34%	70.35%	67.98%
SVM	60.72%	73.52%	61.35%	66.75%	73.65%	77.13%	68.85%
DDC	74.56%	72.71%	75.45%	73.87%	69.91%	58.61%	70.85%
DANN	78.80%	85.27%	81.80%	78.76%	73.72%	64.70%	77.18%
DCTLN	87.98%	85.04%	89.90%	**89.70%**	82.36%	80.60%	85.93%
WDA	**94.89%**	**93.80%**	**97.96%**	88.06%	**88.64%**	**98.96%**	**93.72%**

4.2. CASE II: Ball Screw Fault Diagnosis

In order to further investigate the domain adaptive ability of the proposed method, we set up a test rig for ball screw fault diagnosis, and some vibration signals were collected from the machine, which is shown in Figure 11.

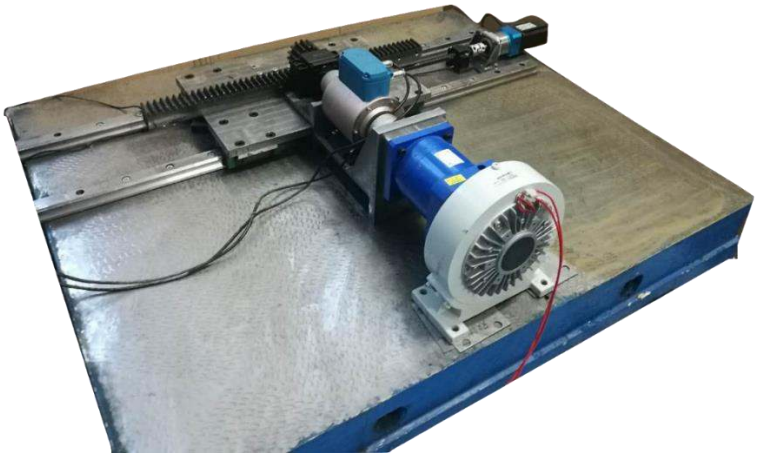


Figure 11. Test rig for ball screw fault diagnosis.

In order to simulate different working conditions, we collected the vibration signals of the ball screw under different forms of end supports. As shown in Table 4, there WERE two ball screw supporting forms: fixed-floating (ζ) and fixed-none (η). In the ζ set, the ball screws were fixed in one end and supported in a floating form on the other end. In the η set, the ball screws were fixed on one end and had no support on the other end.

As seen in the Figure 12, all the transfer learning methods surpassed the traditional methods, showing that transfer learning is essential and effective to bridge the gap between different domains. Moreover, the experiment result shows that WDA had better performance than other state-of-the-art transfer learning methods. Compared to DCTLN, WDA improved about 11.04% more than DCTLN and 21.52% more than DANN. In the condition (' $\zeta \rightarrow \eta$ '), WDA reached an accuracy of 89.13%, greatly surpassing other methods. All this evidence shows that WDA as a transfer learning method has superiority over state-of-the-art methods.

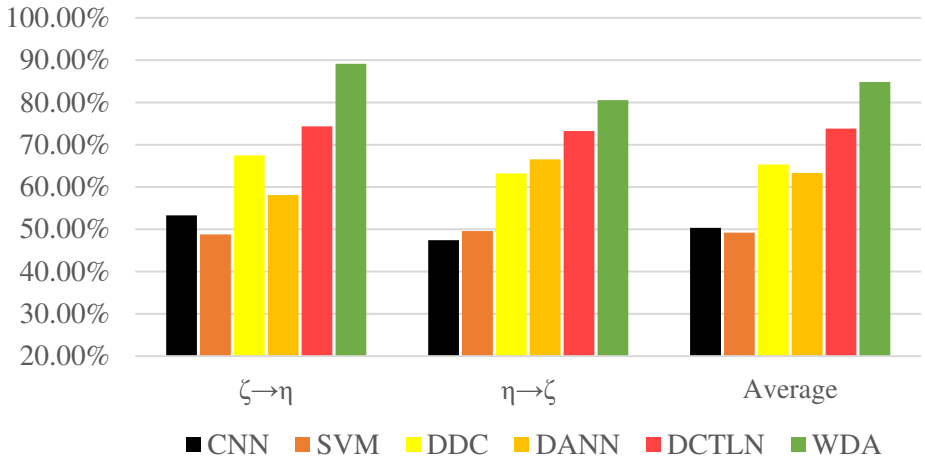


Figure 12. Accuracies of methods under different transfer conditions.

4.3. Feature Visualization

To further investigate the inner mechanism of the proposed method, we applied the feature visualization to output features of CNN from both source and target domains. Different colors represent different features from different health conditions, and the shapes of feature points represent different domains.

In the feature visualization of Figure 13, the features in the whole training process of WDA are shown. Figure 13a–f indicates the visualizations of intermedia feature maps by PCA with the proposed fault diagnosis method from 0 to 100 epochs. The features in the training process gradually gathered into several lines, as in the WDA, cosine similarity was chosen to measure the differences of features from different do-mains. In addition, under the restriction of cosine similarity, features with the same characteristics (e.g., within the same category) turned to keep in the same line rather than a point, although there were some features that did not get in the same line with other features. However, this had little effect on the prediction accuracy of the proposed method. Moreover, we could see that with the training process going on, the source domain features were gradually grouped in several lines with target domain features.

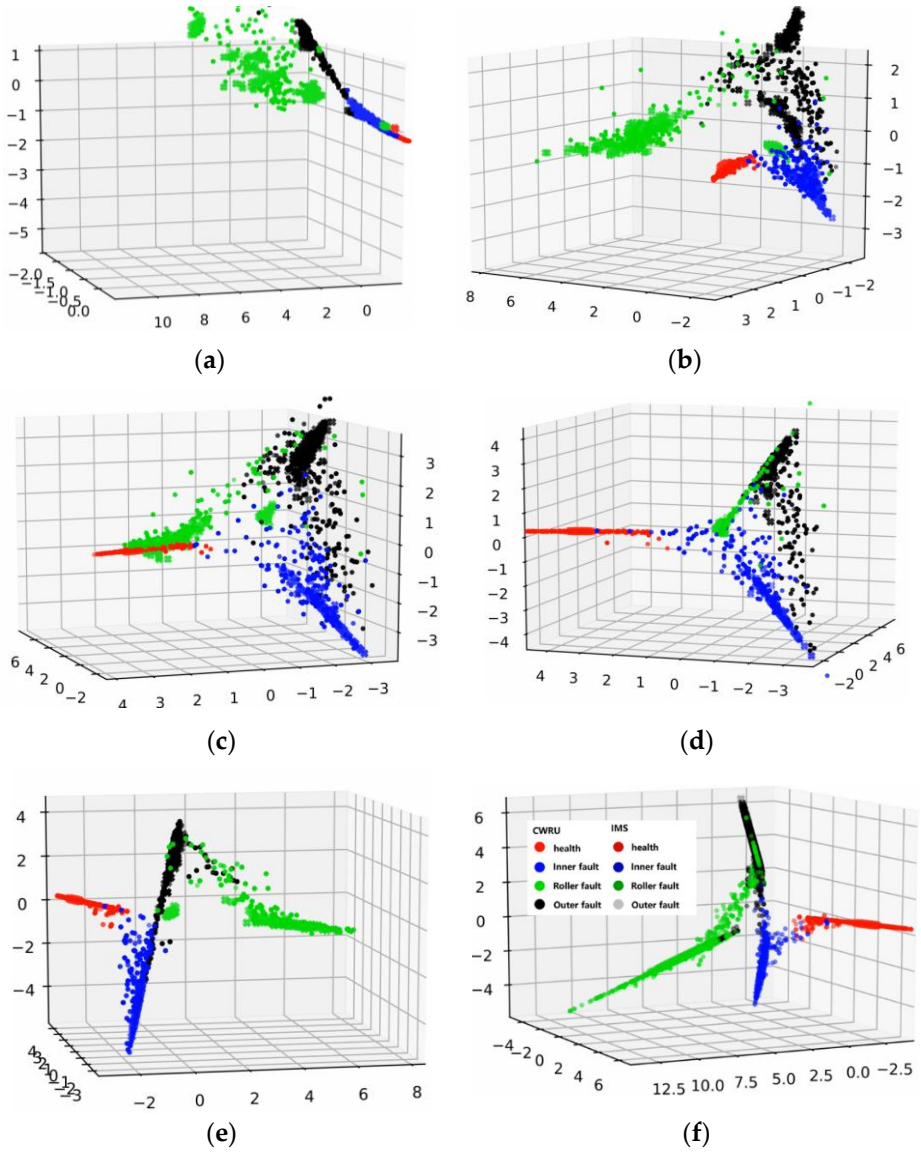


Figure 13. Feature visualization of the training process. (**a**) 0 epochs. (**b**) 20 epochs. (**c**) 40 epochs. (**d**) 60 epochs. (**e**) 80 epochs. (**f**) 100 epochs.

5. The Limitation and Future Works

The proposed method utilized the labeled source domain signals with unlabeled target domain signals for joint training. However, signals from target domains were still class-balanced, which is the limitation of the proposed method. In future works, we will continue to work on the transfer learning task to improve its practicability on the class-unbalanced signals. Moreover, we also want to popularize this method into other transfer learning background problems.

6. Conclusions

In order to produce a more accurate fault diagnosis in unlabeled data, we proposed a Wasserstein distance-based transfer learning fault diagnosis method called WDA. In WDA, the K-M algorithm was introduced to directly calculate the Wasserstein distance. Unlike other methods that use L2-norm measuring the Wasserstein distance, in our methods, cosine similarity was used instead. Moreover, the conception of transfer learning and Wasserstein distance were well explained. Experiments showed that: (1) WDA had better performance than state-of-the-art transfer learning fault diagnosis methods and reached average accuracies of 93.72% and 84.84% on different mechanical parts transfer learning; (2) feature visualization also demonstrated that cosine similarity is efficient to group features from different domains; and (3) the proposed methods could make use of available labeled signals to help unlabeled data classification, thus addressing the problem of the high cost of data labeling and insufficient labeled data. In the age of big data, with the cost of data labeling going up, making use of unlabeled data has become a hot research topic. Thus, transfer learning fault diagnosis requires more attention in research.

Author Contributions: Conceptualization, Z.Z., G.P. and S.L.; methodology, Z.Z., G.P. and S.L.; software, L.W.; validation, Z.Z., G.P. and S.L.; formal analysis, L.W.; investigation, L.W.; resources, L.W. and G.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Voulodimos, A.; Doulamis, N.; Doulamis, A.; Protopapadakis, E. Deep learning for computer vision: A brief review. *Comput. Intell. Neurosci.* **2018**, *2018*, 1–13. [CrossRef] [PubMed]
2. Arik, S.Ö.; Chrzanowski, M.; Coates, A.; Diamos, G.; Gibiansky, A.; Kang, Y.; Li, X.; Miller, J.; Ng, A.; Raiman, J. Deep voice: Real-time neural text-to-speech. In Proceedings of the 34th International Conference on Machine Learning, JMLR. Org, Sydney, Australia, 6–11 August 2017; Volume 70.
3. Angermueller, C.; Pärnamaa, T.; Parts, L.; Stegle, O. Deep learning for computational biology. *Mol. Syst. Biol.* **2016**, *12*, 878. [CrossRef]
4. Zhao, R.; Yan, R.; Chen, Z.; Mao, K.; Wang, P.; Gao, R.X. Deep learning and its applications to machine health monitoring. *Mech. Syst. Signal Process.* **2019**, *115*, 213–237. [CrossRef]
5. Atoui, M.A.; Verron, S.; Kobi, A. Fault Detection and Diagnosis in a Bayesian Network classifier incorporating probabilistic boundary1. *IFAC-PapersOnLine* **2015**, *48*, 670–675. [CrossRef]
6. Rajakarunakaran, S.; Venkumar, P.; Devaraj, D.; Rao, K.S.P. Artificial neural network approach for fault detection in rotary system. *Appl. Soft Comput.* **2008**, *8*, 740–748. [CrossRef]
7. Portnoy, I.; Melendez, K.; Pinzon, H.; Sanjuan, M. An improved weighted recursive PCA algorithm for adaptive fault detection. *Control Eng. Pract.* **2016**, *50*, 69–83. [CrossRef]
8. Lu, C.; Wang, Z.-Y.; Qin, W.-L.; Ma, J. Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification. *Signal Process.* **2017**, *130*, 377–388. [CrossRef]
9. Li, C.; Zhang, W.; Peng, G.; Liu, S. Bearing fault diagnosis using fully-connected winner-take-all autoencoder. *IEEE Access* **2017**, *6*, 6103–6115. [CrossRef]
10. Zarei, J. Induction motors bearing fault detection using pattern recognition techniques. *Expert Syst. Appl.* **2012**, *39*, 68–73. [CrossRef]
11. Janssens, O.; Van de Walle, R.; Loccufier, M.; Van Hoecke, S. Deep learning for infrared thermal image based machine health monitoring. *Ieee/Asme Trans. Mechatron.* **2017**, *23*, 151–159. [CrossRef]
12. Zhang, B.; Zhang, S.; Li, W. Bearing performance degradation assessment using long short-term memory recurrent network. *Comput. Ind.* **2019**, *106*, 14–29. [CrossRef]

13. Cui, L.; Wang, X.; Xu, Y.; Jiang, H.; Zhou, J. A novel Switching Unscented Kalman Filter method for remaining useful life prediction of rolling bearing. *Measurement* **2019**, *135*, 678–684. [CrossRef]
14. Zhang, W.; Li, C.; Peng, G.; Chen, Y.; Zhang, Z. A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mech. Syst. Signal Process.* **2018**, *100*, 439–453. [CrossRef]
15. Zhu, Z.; Peng, G.; Chen, Y.; Gao, H. A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis. *Neurocomputing* **2019**, *323*, 62–75. [CrossRef]
16. Lu, W.; Liang, B.; Cheng, Y.; Meng, D.; Yang, J.; Zhang, T. Deep model based domain adaptation for fault diagnosis. *IEEE Trans. Ind. Electron.* **2016**, *64*, 2296–2305. [CrossRef]
17. Wen, L.; Li, X.; Gao, L.; Zhang, Y. A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Trans. Ind. Electron.* **2017**, *65*, 5990–5998. [CrossRef]
18. Xie, J.; Zhang, L.; Duan, L.; Wang, J. On cross-domain feature fusion in gearbox fault diagnosis under various operating conditions based on transfer component analysis. In Proceedings of the 2016 IEEE International Conference on Prognostics and Health Management (ICPHM), Ottawa, ON, Canada, 20–22 June 2016.
19. Guo, L.; Lei, Y.; Xing, S.; Yan, T.; Li, N. Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data. *IEEE Trans. Ind. Electron.* **2018**, *66*, 7316–7325. [CrossRef]
20. Udmale, S.S.; Singh, S.K.; Singh, R.; Sangaiah, A.K. Multi-Fault Bearing Classification Using Sensors and ConvNet-Based Transfer Learning Approach. *IEEE Sens. J.* **2020**, *20*, 1433–1444. [CrossRef]
21. Hasan, M.J.; Islam, M.M.; Kim, J.-M. Acoustic spectral imaging and transfer learning for reliable bearing fault diagnosis under variable speed conditions. *Measurement* **2019**, *138*, 620–631. [CrossRef]
22. Zhang, W.; Li, X.; Ma, H.; Luo, Z.; Li, X. Universal Domain Adaptation in Fault Diagnostics with Hybrid Weighted Deep Adversarial Learning. *IEEE Trans. Ind. Informatics* **2021**. [CrossRef]
23. Zhang, W.; Li, X.; Ma, H.; Luo, Z.; Li, X. Federated learning for machinery fault diagnosis with dynamic validation and self-supervision. *Knowl.-Based Syst.* **2021**, *213*, 106679. [CrossRef]
24. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein Generative Adversarial Networks. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 214–223.
25. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. Improved training of wasserstein gans. *arXiv* **2017**, arXiv:1704.00028.
26. Zhang, M.; Wang, D.; Lu, W.; Yang, J.; Li, Z.; Liang, B. A Deep Transfer Model With Wasserstein Distance Guided Multi-Adversarial Networks for Bearing Fault Diagnosis Under Different Working Conditions. *IEEE Access* **2019**, *7*, 65303–65318. [CrossRef]
27. Shen, J.; Qu, Y.; Zhang, W.; Yu, Y. Wasserstein distance guided representation learning for domain adaptation. *arXiv* **2017**, arXiv:1707.01217.
28. Cheng, C.; Zhou, B.; Ma, G.; Wu, D.; Yuan, Y. Wasserstein Distance based Deep Adversarial Transfer Learning for Intelligent Fault Diagnosis. *arXiv* **2019**, arXiv:1903.06753.
29. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [CrossRef]
30. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2018.
31. Long, M.; Wang, J.; Ding, G.; Sun, J.; Yu, P.S. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2013.
32. Shen, F.; Chen, C.; Yan, R.; Gao, R.X. Bearing fault diagnosis based on SVD feature extraction and transfer learning classification. In Proceedings of the 2015 Prognostics and System Health Management Conference (PHM), Beijing, China, 21–23 October 2015.
33. Tong, Z.; Li, W.; Zhang, B.; Jiang, F.; Zhou, G. Bearing fault diagnosis under variable working conditions based on domain adaptation using feature transfer learning. *IEEE Access* **2018**, *6*, 76187–76197. [CrossRef]
34. Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep domain confusion: Maximizing for domain invariance. *arXiv* **2014**, arXiv:1412.3474.
35. Yan, H.; Ding, Y.; Li, P.; Wang, Q.; Xu, Y.; Zuo, W. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2017.
36. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **2016**, *17*, 2030–2096.
37. Bang, D.; Shim, H. Improved training of generative adversarial networks using representative features. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 433–442.
38. Wiatrak, M.; Albrecht, S.V. Stabilizing Generative Adversarial Network Training: A Survey. 2019. Available online: https://arxiv.org/abs/1910.00927v2 (accessed on 26 June 2021).
39. Berthelot, D.; Schumm, T.; Metz, L. Began: Boundary equilibrium generative adversarial networks. *arXiv* **2017**, arXiv:1703.10717.
40. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [CrossRef]
41. Liu, Z.-Y.; Qiao, H.; Xu, L. An extended path following algorithm for graph-matching problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1451–1456. [CrossRef]

42. Cui, H.; Zhang, J.; Cui, C.; Chen, Q. Solving large-scale assignment problems by Kuhn-Munkres algorithm Hong Cui1, Jingjing Zhang2, b, Chunfeng Cui3, c, Qinyu Chen4, d. 2016. Available online: https://www.atlantis-press.com/proceedings/ameii-16/25854748 (accessed on 26 June 2021).
43. Carpaneto, G.; Toth, P. Algorithm 548: Solution of the assignment problem [H]. *ACM Trans. Math. Softw. (TOMS)* **1980**, *6*, 104–111. [CrossRef]
44. Berenguer, J.A.M.; Coello, C.A.C. Evolutionary many-objective optimization based on kuhn-munkres' algorithm. In *International Conference on Evolutionary Multi-Criterion Optimization*; Springer: Cham, Switzerland, 2015; pp. 3–17.
45. Zhu, H.; Zhou, M. Efficient role transfer based on Kuhn–Munkres algorithm. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2011**, *42*, 491–496. [CrossRef]
46. Qiu, H.; Lee, J.; Lin, J.; Yu, G. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. *J. Sound Vib.* **2006**, *289*, 1066–1090. [CrossRef]
47. Lou, X.; Loparo, K.A. Bearing fault diagnosis based on wavelet transform and fuzzy inference. *Mech. Syst. Signal Process.* **2004**, *18*, 1077–1095. [CrossRef]
48. Suykens, J.A.; Van Gestel, T.; De Brabanter, J. *Least Squares Support Vector Machines*; World Scientific: Singapore, 2002. Available online: https://doi.org/10.1142/5089 (accessed on 26 June 2021).