

## Article

# A Transfer Learning Framework with a One-Dimensional Deep Subdomain Adaptation Network for Bearing Fault Diagnosis under Different Working Conditions

Ruixin Zhang<sup>1</sup> and Yu Gu<sup>2,3,4,\*</sup>

- <sup>1</sup> College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China; rxzhang1@mail.buct.edu.cn
- <sup>2</sup> Guangdong Province Key Laboratory of Petrochemical Equipment Fault Diagnosis, Maoming 525000, China
- <sup>3</sup> Beijing Advanced Innovation Center for Soft Matter Science and Engineering, Beijing University of Chemical Technology, Beijing 100029, China
- <sup>4</sup> Department of Chemistry, Institute of Inorganic and Analytical Chemistry, Goethe-University, Max-von-Laue-Str. 9, 60438 Frankfurt, Germany
- \* Correspondence: guyu@mail.buct.edu.cn

**Abstract:** Accurate and fast rolling bearing fault diagnosis is required for the normal operation of rotating machinery and equipment. Although deep learning methods have achieved excellent results for rolling bearing fault diagnosis, the performance of most methods declines sharply when the working conditions change. To address this issue, we propose a one-dimensional lightweight deep subdomain adaptation network (1D-LDSAN) for faster and more accurate rolling bearing fault diagnosis. The framework uses a one-dimensional lightweight convolutional neural network backbone for the rapid extraction of advanced features from raw vibration signals. The local maximum mean discrepancy (LMMMD) is employed to match the probability distribution between the source domain and the target domain data, and a fully connected neural network is used to identify the fault classes. Bearing data from the Case Western Reserve University (CWRU) datasets were used to validate the performance of the proposed framework under different working conditions. The experimental results show that the classification accuracy for 12 tasks was higher for the 1D-LDSAN than for mainstream transfer learning methods. Moreover, the proposed framework provides satisfactory results when a small proportion of the unlabeled target domain data is used for training.



**Citation:** Zhang, R.; Gu, Y. A Transfer Learning Framework with a One-Dimensional Deep Subdomain Adaptation Network for Bearing Fault Diagnosis under Different Working Conditions. *Sensors* **2022**, *22*, 1624. <https://doi.org/10.3390/s22041624>

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

Received: 28 January 2022  
Accepted: 16 February 2022  
Published: 18 February 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** fault diagnosis; deep learning; rolling bearing; domain adaptation; transfer learning

## 1. Introduction

Due to advances in industrial technology, rotating machinery is increasingly used in many fields, such as electric power generation, chemical production, and aerospace [1,2]. Rolling bearings are indispensable elements in rotating machines [3] and are the main source of faults in this equipment [4]. Rotating machines may operate under unfavorable conditions, such as high ambient temperatures, high humidity, and overload conditions, resulting in bearing malfunctions [5]. Bearing faults can cause significant damage to mechanical equipment [6]. Therefore, accurate and rapid methods for rolling bearing fault diagnosis are required to ensure the normal operation of rotating machinery.

In recent years, artificial intelligence methods, such as heuristic algorithm [7], expert knowledge-based methods [8], and deep learning (DL) models [9], have gained increasing attention in diverse fields. In particular, DL models have been broadly employed for machinery fault detection and diagnosis systems [10]. Most DL models, such as the long short-term memory network (LSTM) [11], deep belief network (DBN) [12], and convolutional neural network (CNN) [13–15], perform well if the datasets of the source domain and target domain tasks have the same distribution [16]. However, this assumption is

rarely applicable in practical conditions. In many real-world applications, the working conditions during testing and training differ [17]. Therefore, the unlabeled testing data may not have the same distribution as the labeled training data, potentially leading to the misclassification of DL methods [18]. Thus, it is essential to consider the change in working conditions to improve the accuracy and efficiency of bearing fault diagnosis.

Transfer learning aims to extract information from one or more source tasks and apply it to a target task [19]. Deep domain adaptation (DDA), a branch of transfer learning, is designed to train a classifier or other predictor when the source domain data and target domain data have different distributions [20]. Since DDA can minimize the distribution discrepancy between different domains, it is well suited for solving cross-domain diagnosis tasks. Yang et al. [21] developed a bearing fault diagnosis framework based on a two-dimensional CNN and DDA. In this framework, multikernel maximum mean discrepancy (MK-MMD) was used for domain adaptation in four convolution layers. Although the method achieved an average accuracy value of 99.14% for 12 transfer learning tasks, the diagnostic accuracy was only 97.52% when substantial differences in working conditions existed. Wu et al. [22] converted raw data into two-dimensional time–frequency images using continuous wavelet transform (CWT) and proposed an accurate model consisting of a CNN and a deep adaptation network (DAN) for bearing fault diagnosis. The framework achieved a diagnostic accuracy score of more than 98% on the bearing fault dataset of Case Western Reserve University (CWRU). Although the method achieved satisfactory accuracy for transfer learning tasks, converting the vibration signal into images was computationally complex and time-consuming. Zhang et al. [23] proposed a domain adaptation framework using an adversarial learning strategy for machinery fault diagnostics. An instance-level weighted mechanism was also integrated to address the open-set problem. Jiao et al. [24] proposed a residual network to extract features from raw vibration data and combined the maximum mean discrepancy (MMD) with a domain adversarial strategy to align the domain distribution. The method obtained an average fault diagnosis accuracy value of 99.32% for 12 transfer learning tasks on the CWRU dataset. However, domain adversarial-based methods contain several loss functions and converge slowly. Although DDA approaches have been utilized for fault diagnosis, most methods (mapping-based and adversarial-based methods) assume that the global distribution differs for the target and source domains and try to reduce this difference. However, differences in the subdomain distribution of features and output labels among different working conditions are rarely considered. Unsatisfactory results could occur if the fine-grained information is not captured [25]. Therefore, a subdomain adaptation strategy that can exploit the local affinity to capture the fine-grained information of each category is incorporated to match the subdomain distributions of data from different working conditions.

In this paper, a novel one-dimensional lightweight deep subdomain adaptation network (1D-LDSAN) framework is proposed for bearing fault diagnosis under different working conditions. A 1D-CNN backbone is used to extract sufficient features from the raw data as input to a fully connected (FC) classifier, which diagnoses the faults accurately by utilizing advanced data features. A subdomain adaptation strategy is employed to match the subdomain distributions of data from different working conditions. The contributions of this paper are summarized as follows.

- (1) A novel fault diagnosis framework (1D-LDSAN) consisting of a feature extraction module and a classification and adaptation module is proposed. The feature extraction module, a lightweight 1D-CNN backbone, is designed to extract a sufficient number of comprehensive and significant features of different faults from the raw vibration signal. The classification and adaptation module is used to classify the data and minimize the subdomain distribution discrepancy of the data from two domains to improve the classification performance.

- (2) Comparative experiments are performed to verify the performance of the proposed framework on the CWRU dataset. Five other approaches, including deep domain confusion (DDC), a domain-adversarial neural network (DANN), a residual joint adaptation

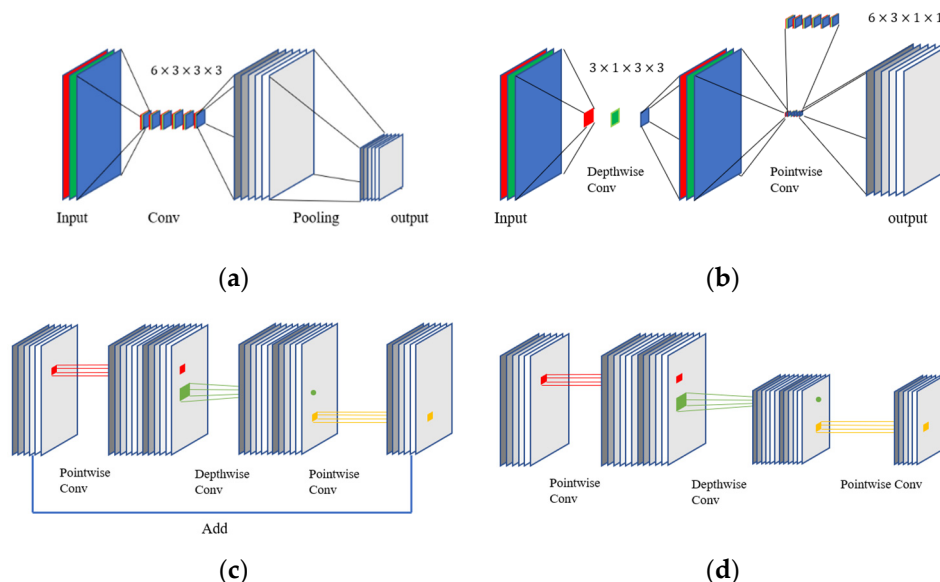
adversarial network (RJANN), a Wasserstein distance-guided multi-adversarial network (WDMAN), and a one-dimensional CNN, were evaluated for comparison to assess the performance of the 1D-LDSAN. The results demonstrate the effectiveness and superiority of the proposed framework.

The remaining parts of this paper are organized as follows. The theoretical background, the CNN, the domain adaptation, and MMD are introduced in Section 2. In Section 3, the details of the proposed 1D-LDSAN model are presented. The datasets, the experimental results, and the discussion are provided in Section 4. Finally, the conclusions are summarized in Section 5.

## 2. Related Works

### 2.1. Convolutional Neural Network

A CNN is a multi-stage neural network composed of convolutional blocks and FC layers [26]. Traditionally, a convolutional block is composed of a convolution layer and a pooling layer [27], as shown in Figure 1a. In general, a batch normalization (BN) layer [28] is added after the convolution layer to improve the network training speed, prevent overfitting, and control gradient explosion and gradient disappearance. An activation function is required for nonlinear transformation after the convolution operation. The purpose of the activation function is to add nonlinear factors to the feature map after the convolution operation [29]. In this paper, the rectified linear unit (ReLU) [30] activation function is selected. Traditional CNNs require a pooling layer after activation to adjust the output of the convolution layer [31]. Many techniques have been used recently to replace the pooling function. MobileNet V2 [32] uses step convolution to replace the pooling layer. The classifier of a CNN is often an FC layer [33] that maps the learned features to the sample label space.



**Figure 1.** Comparison of convolutional blocks for different architectures. (a) Convolution pooling block; (b) separable block; (c) inverted residual block; (d) inverted residual block (stride).

### 2.2. Domain Adaptation

Domain adaptation is a specific area of transfer learning; it refers to training a discriminative model in the presence of a domain shift between domains [34]. Domain adaptation establishes a knowledge transfer from the labeled source domain to the unlabeled target domain by using domain-invariant structures that bridge different domains with substantial discrepancies in the distribution [35,36].

In real-world applications, the working conditions of machines are often changed. Different working conditions are defined as different domains. The working condition with

labeled data is defined as the source domain  $D_s = \{x_i, y_i\}_{i=1}^n$ , and the working condition with unlabeled data is defined as the target domain  $D_t = \{x_j\}_{j=1}^m$ . It is assumed that they have the same feature space,  $X_s = X_t$ , and category space,  $Y_s = Y_t$ . However, the distributions of the two domains,  $D_s$  and  $D_t$ , are different,  $P_s(x_s) \neq P_t(x_t)$ . The goal of this work is to use labeled source domain data  $D_s$  and unlabeled target domain data  $D_t$  to learn a classifier  $f: x_t \rightarrow y_t$  to predict the labels  $y_t \in Y_t$  of the target domain data  $D_t$ .

### 2.3. MobileNet V2

MobileNet V2 [32] is a lightweight CNN model for image processing. The main block in the model is inherited from the separable block [37], as shown in Figure 1b, and its main structure is combined with the residual structure [38] to construct an inverted residual block with a linear bottleneck. MobileNet V2 is created by embedding the inverted residual block instead of a standard convolution layer, as shown in Figure 1c,d. The linear bottleneck removes nonlinearities in the narrow layers because they destroy information in low-dimensional space. Hence, the linear bottleneck retains the representativeness of the model [39].

### 2.4. Local Maximum Mean Discrepancy (LMMD)

The MMD [40] calculates the discrepancy between two distributions by mapping sample points to the reproducing kernel Hilbert space (RKHS), which is a kernel method. Minimizing the MMD between the two domains aligns the edge probability distribution between the two domains in a neural network. The MMD between the source domain  $D_s$  and target domain  $D_t$  is defined as

$$\begin{aligned} \text{MMD}^2(D_s, D_t) &= \left\| \frac{1}{n_s} \sum_{x_i \in D_s} \varnothing(x_i) - \frac{1}{n_t} \sum_{x_j \in D_t} \varnothing(x_j) \right\|_H^2 \\ &= \frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} k(x_i^s, x_j^s) + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} k(x_i^t, x_j^t) - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} k(x_i^s, x_j^t) \end{aligned} \quad (1)$$

where  $H$  represents the RKHS,  $\varnothing(\cdot)$  denotes the feature map to map the raw sample to the RKHS,  $k$  is the kernel function  $k(x^s, x^t) = \langle \varnothing(x^s), \varnothing(x^t) \rangle$ , and  $\langle \cdot, \cdot \rangle$  represents the inner product of two vectors.

The local MMD (LMMD) [25] is an improved version of the MMD for matching the local probability distribution. The LMMD between the source domain data  $D_s$  and target domain data  $D_t$  is calculated as follows:

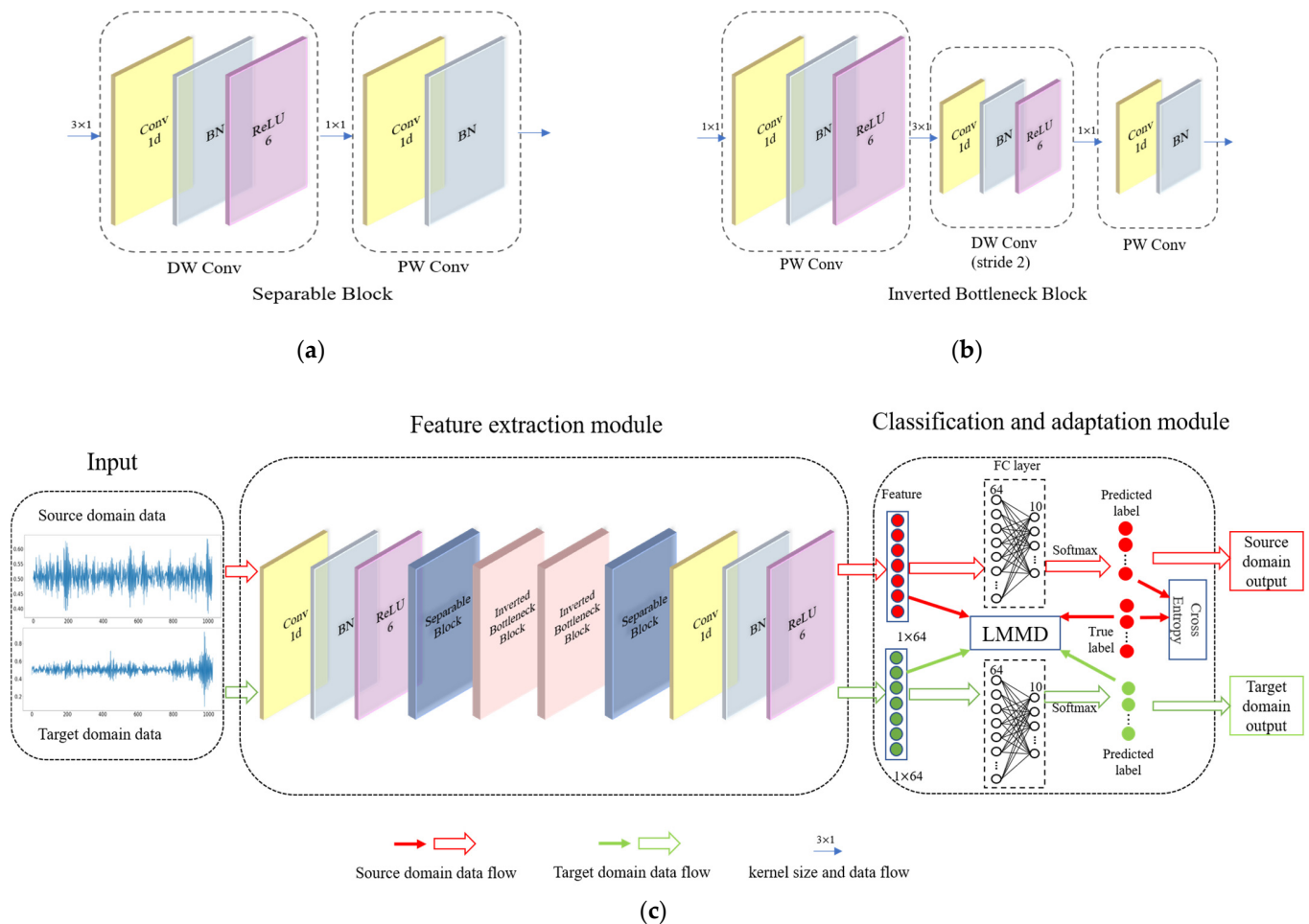
$$\begin{aligned} \text{LMMD}^2(D_s, D_t) &= \frac{1}{C} \sum_{c=1}^C \left\| \sum_{x_i^s \in D_s} \omega_i^{sc} \varnothing(x_i^s) - \sum_{x_j^t \in D_t} \omega_j^{tc} \varnothing(x_j^t) \right\|_H^2 \\ &= \frac{1}{C} \sum_{c=1}^C \left[ \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} \omega_i^{sc} \omega_j^{sc} k(x_i^s, x_j^s) + \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} \omega_i^{tc} \omega_j^{tc} k(x_i^t, x_j^t) - 2 \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \omega_i^{sc} \omega_j^{tc} k(x_i^s, x_j^t) \right] \end{aligned} \quad (2)$$

where  $x_i^s$  and  $x_j^t$  represent the feature map of the source domain data and target domain data, respectively;  $\omega_i^{sc}$  and  $\omega_j^{tc}$  denote the weights of  $x_i^s$  and  $x_j^t$  belonging to class  $c$ ,  $\sum_{i=1}^{n_s} \omega_i^{sc}$  and  $\sum_{j=1}^{n_t} \omega_j^{tc}$  are equal to 1, and  $\sum_{x_i \in D} \omega_i^c \varnothing(x_i)$  is the weighted sum of class  $C$ .

## 3. Materials and Methods

Although DDA approaches have been used for fault diagnosis, most existing methods assume that the global distribution differs for the target and source domains and try to reduce this difference. However, differences in the subdomain distribution of features and output labels among different working conditions are rarely considered. Thus, fine-grained information in the categories may not be detected. Therefore, the 1D-LDSAN framework is proposed for the fault diagnosis of rolling bearings under different working conditions.

As shown in Figure 2c, the proposed framework consists of a feature extraction module and a classification and adaptation module. The details will be introduced in the following subsections.



**Figure 2.** Schematic diagram of the proposed 1D-LDSAN framework. (a) is the structure diagram of the separable block shown in (c). (b) presents the structure diagram of the inverted bottleneck block shown in (c). (c) displays the overall framework.

### 3.1. Framework Structure

#### 3.1.1. Feature Extraction Module

Inspired by MobileNet V2, we designed the feature extraction module to extract deep features from the raw data from the source domain and target domain. As shown in Figure 2c, the feature extraction module consists of two regular convolutional blocks and four unique convolutional blocks. The input size of the feature extraction module is  $1024 \times 1$ . The regular convolutional block consists of a convolutional layer, a BN layer, and a ReLU6 layer. The first regular convolutional block has a kernel size of  $4 \times 1$  and a stride of 4 to reduce the length of input data. The second regular convolutional block has a kernel size of  $1 \times 1$  and a stride of 1 to expand the number of feature channels. The unique convolutional blocks consist of two types: a separable block and an inverted bottleneck block, as shown in Figure 2a,b. The separable block is divided into two layers. The first layer is a depthwise convolution that performs lightweight filtering by applying a single convolutional filter per input channel. The second layer is a  $1 \times 1$  convolution (pointwise convolution) responsible for creating new features by computing linear combinations of the input channels. The two-layer operation of the separable block replaces the full convolutional operator, which substantially reduces the number of parameters of the convolution kernel. In the inverted



bottleneck block, a pointwise convolution is inserted in front of the separable convolution layer. In the second layer, the stride of the depthwise separable convolution is two. The structure maps the features to a high-dimensional space for fine-grained feature extraction. The details of the CNN backbone are listed in Table 1.

**Table 1.** Details of feature extraction module.

Block	Layer	Parameters	Output Size
Input	Input	/	1024 × 1
Regular Conv	ConvBNReLU6	Kernel size = 6@4 × 1 × 1 stride = 4	256 × 6
Separable Block	ConvBNReLU6	Kernel size = 6@3 × 1 stride = 1	256 × 6
	ConvBN	Kernel size = 16@1 × 1v4 stride = 1	256 × 16
Inverted Bottleneck Block	ConvBNReLU6	Kernel size = 96@1 × 1 × 16 stride = 1	256 × 96
	ConvBNReLU6	Kernel size = 96@3 × 1 stride = 2	128 × 96
	ConvBN	Kernel size = 24@1 × 1 × 96 stride = 1	128 × 24
Inverted Bottleneck Block	ConvBNReLU6	Kernel size = 144@1 × 1 × 24 stride = 1	128 × 144
	ConvBNReLU6	Kernel size = 144@3 × 1 stride = 2	64 × 144
	ConvBN	Kernel size = 32@1 × 1 × 144 stride = 1	64 × 32
Separable Block	ConvBNReLU6	Kernel size = 32@3 × 1 stride = 1	64 × 32
	ConvBN	Kernel size = 48@1 × 1 × 32 stride = 1	64 × 48
Regular Conv	ConvBNReLU6	Kernel size = 64@1 × 1 × 48 stride = 1	64 × 64
Avg Pooling	/	/	1 × 64

### 3.1.2. Classification and Adaptation Module

As shown in Figure 2c, the classifier of the framework is an FC neural network. The weights of the classifier are shared by the source domain features and the target domain features. The number of neurons in the classifier is the same as the number of extracted features. The input of the LMMD function has four items, including the source domain features, the target domain features, the true label of the source domain data, and the predicted label of the target domain data. The classification and adaptation module is designed to minimize the classification errors of the source domain using a cross-entropy function and reduce the subdomain distribution discrepancy between the target domain and the source domain using the LMMD.

### 3.2. Optimization Objectives

This subsection describes the optimization objectives of the proposed framework. The framework has two optimization objectives, as shown in Figure 2c. The cross-entropy [41] function is implemented to minimize the classification error of the source domain dataset; it is defined as follows:

$$L_c = - \sum_{c=1}^C y_{sk} \times \log \frac{e^{\tilde{y}_{sc}}}{\sum_j e^{\tilde{y}_{sj}}} \quad (3)$$

where  $\tilde{y}_s$  is the predicted label vector of the source domain data,  $y_s$  is the true label vector of the source domain data, and  $C$  is the number of labels.

As described in Section 2.3, the LMMD is used to minimize the local subdomain distribution between the source domain data and target domain data. During training, the LMMD loss  $L_{LMMD}$  is calculated as follows:

$$L_{LMMD} = LMMD^2(x_s, x_t, y_s, \tilde{y}_t) \quad (4)$$

where  $x_s$  and  $x_t$  are the features extracted from the source domain data and the target domain data, respectively.  $\tilde{y}_t$  is the label vector of the target domain data predicted by the framework.  $y_s$  is the true label vector of the source domain data.

The cross-entropy function and loss function are combined and represent the optimization goal, which is described as

$$L = L_c + \lambda L_{LMMD} \quad (5)$$

where  $\lambda$  is the tradeoff parameter.

### 3.3. Network Training Strategy

Some DDA methods [18,42] must pre-train the neural network with the source domain data, which increases the training time and complicates the training process. This paper uses a strategy [20] of gradually increasing the tradeoff parameter  $\lambda$  from 0 to 0.2 during training, where  $\lambda = 0.2 \times \left(2 / \left(1 + e^{(-10 \times (epoch) / epochs)}\right) - 1\right)$ . An Adam [43] optimization strategy is used to optimize the network parameters, and a data augmentation algorithm is implemented to enhance network generalization. The data are jittered up and down randomly during training. Exponential attenuation of the learning rate is implemented to improve the stability of the framework during training. The learning rate has an initial value of 0.01 and decreases with an increase in the number of training epochs. The batch size is 64. During training, 80% of the source domain data and 50% of the unlabeled target domain data are used for domain-adaptive training. The remaining source data are used for validation, and the remaining target domain data are used for testing. The pipeline of training the 1D-LDSAN is presented in Algorithm 1.

---

#### Algorithm 1 1D-LDSAN.

---

**Input:** labeled source domain data and unlabeled target domain data.

**Output:** predicted category of target domain.

**Begin**

Step 1: normalize source domain and target domain data

Step 2: initial neural network parameters with random values

Step 3: input the normalized source domain and target domain data into the neural network to calculate  $L_c$  and  $L_{LMMD}$

Step 4: optimize the parameters of neural network using Adam strategy, repeat Step 3 and Step 4 until the specified epoch is reached

Step 5: save the model

Step 6: diagnose the target domain data using the trained model

Step 7: output the classification results

**End**

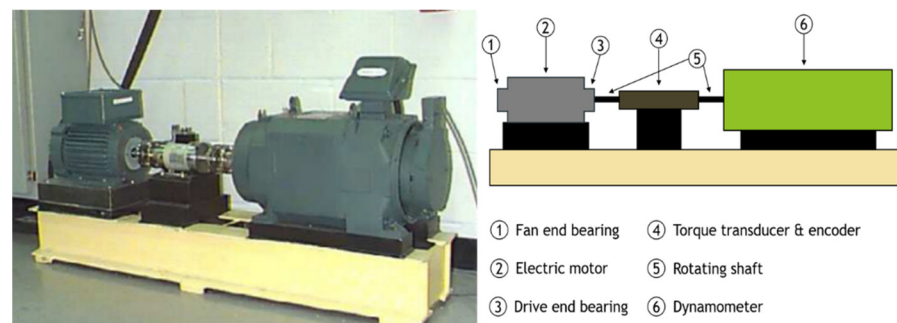
---

## 4. Experiments

In this study, the CWRU dataset [44] was used to evaluate the performance and practicability of the proposed 1D-LDSAN framework. Five other methods were evaluated for comparison. Pytorch 1.8.1 was used to implement the proposed framework, and a computer with the Windows 10 operating system and a gtx1050 GPU was used.

### 4.1. Dataset Description

The CWRU dataset is a standard bearing fault dataset collected by the bearing center at CWRU. It is commonly used to validate and/or improve motor condition assessment techniques. Here, it was used to verify the performance of the framework. Figure 3 shows a photo and a diagram of the experimental platform. Bearings are used at the fan end and the drive end of the motor to enable the rotation of the motor's shaft. The drive-end bearing is an SKF6205 deep groove ball bearing, and the fan-end bearing is an SKF6203 deep groove ball bearing. Two acceleration sensors were placed above the bearing pedestal at the fan end and drive end of the motor, respectively, to collect the vibration acceleration signal of the faulty bearing.



**Figure 3.** Photo (left) and schematic diagram (right) of the experimental device at Case Western Reserve University to assess bearing failure [44].

There are three fault types in this dataset, i.e., inner race fault (IF), outer race fault (OF), and roller fault (RF). The faults are machined by an electrical discharge machine (EDM), and each fault type has three damage sizes (0.007, 0.014, and 0.021 inches). Therefore, the CWRU dataset has ten classes (one normal class and nine fault classes (3 fault classes  $\times$  3 fault diameters)). The fault data were collected under four operating conditions in the experiment, including 0 HP, 1 HP, 2 HP, and 3 HP, with a sampling frequency of 12 kHz. Thus, the data were divided into four domains (A, B, C, and D), and there were 12 transfer learning tasks. The details of the dataset are presented in Table 2. Each sample contained 1024 data points.

**Table 2.** Description of the CWRU dataset.

Domain	Load (HP)	Rotating Speed (r/min)	Number of Samples	Number of Labels
A	0	1797	1186	10
B	1	1772	1186	10
C	2	1750	1185	10
D	3	1730	1189	10

#### 4.2. Comparison of Different Signal Lengths

We extracted features from the raw bearing data using four signal lengths (256, 512, 1024, and 2048) to determine the optimum signal length for the 1D-LDSAN framework. Table 3 lists the number of samples obtained using the four signal lengths. In the experiment, the batch sizes were 256, 128, 64, and 32, respectively. A total of 12 transfer learning tasks were conducted. For example, transfer task A–B indicates that 0 HP is the source domain and 1 HP is the target domain.

**Table 3.** Description of the samples of the four signal lengths.

Domain	256 Points	512 Points	1024 Points	2048 Points
A	4763	2379	1186	591
B	4762	2379	1186	591
C	4760	2377	1185	591
D	4769	2383	1189	592

The results are listed in Table 4. It was found that 1024 provided the best results and was used as the signal length. This experiment demonstrates the power of the proposed framework to model fault-related nonlinear vibration signals.



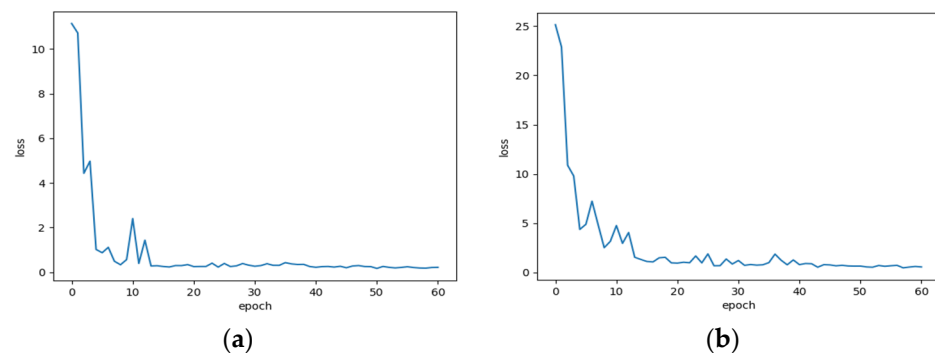
**Table 4.** Classification accuracies of different signal lengths. The numbers in bold indicate the highest classification accuracy in corresponding task.

Task	256 Points	512 Points	1024 Points	2048 Points
A-B	98.84%	99.65%	99.90%	<b>99.93%</b>
A-C	97.04%	99.45%	99.89%	<b>99.90%</b>
A-D	97.44%	99.64%	<b>99.98%</b>	99.90%
B-A	98.94%	99.79%	<b>99.96%</b>	98.70%
B-C	99.11%	99.66%	<b>100.00%</b>	99.93%
B-D	95.96%	96.27%	<b>99.97%</b>	99.93%
C-A	97.95%	98.90%	<b>99.77%</b>	98.80%
C-B	98.44%	99.45%	99.55%	<b>99.73%</b>
C-D	98.70%	99.80%	99.93%	<b>99.97%</b>
D-A	94.16%	97.11%	<b>99.54%</b>	98.80%
D-B	92.42%	95.94%	<b>99.48%</b>	97.17%
D-C	98.28%	99.19%	<b>99.89%</b>	99.77%
AVG	97.27%	98.74%	<b>99.82%</b>	99.38%

#### 4.3. Comparison with Other Transfer Learning Methods

The detection accuracy of 1D-LDSAN was compared with that of five other methods, including a 1D-CNN, DDC [45], DANN [46], RJANN [24], and the WDMAN [47]. The 1D-CNN has the same architecture as the proposed 1D-LDSAN for feature extraction. It uses only the source domain samples to train a domain-shared 1D-CNN, and the model is tested with the target domain samples. The DDC is a mapping-based DDA method. DANN is a type of adversarial neural network. In this experiment, the 1D-CNN was employed as the feature extractor of the DDC and DANN. Each method was implemented with the optimal parameters. RJANN and WDMAN are other popular DDA methods used for fault diagnosis.

Figure 4 displays the 1D-LDSAN's validation loss during the training process. The convergence occurs after 50 epochs. Each experiment was repeated ten times for each model. The average detection results on the CWRU dataset are summarized in Table 5. The proposed 1D-LDSAN achieves an average detection accuracy score of 99.82%, outperforming the other five methods. Among the five comparison methods, the four deep transfer learning methods are superior to the DL method. Although the WDMAN and RJANN achieve average classification accuracies above 99%, there are gaps between them and the proposed method for some transfer tasks. The 1D-CNN, DDC, and DANN achieve good results when the discrepancy is relatively small, such as the transfer task between A and B. However, the transfer task performance is unsatisfactory for the three methods when the working conditions change dramatically, resulting in low fault diagnosis accuracy. Notably, the proposed 1D-LDSAN exhibits excellent accuracy for all 12 transfer tasks. Therefore, these results demonstrate the effectiveness and superiority of the proposed method.

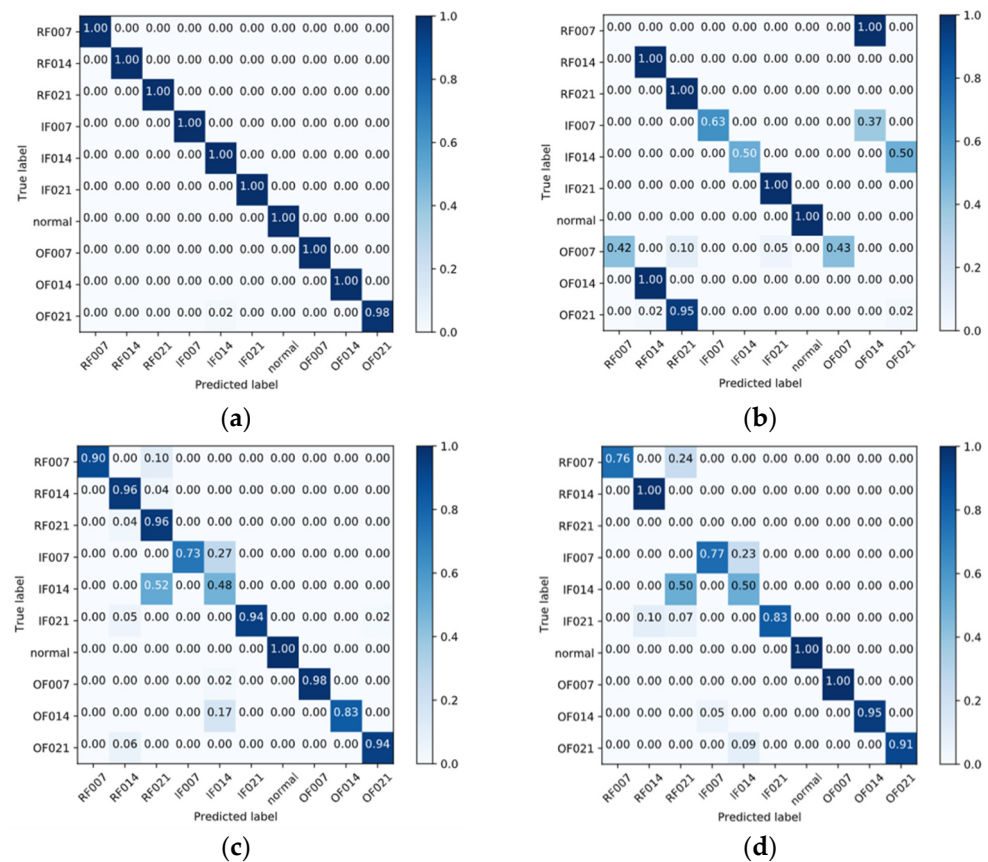


**Figure 4.** The validation loss of the 1D-LDSAN during the training on the CWRU dataset. (a) Source domain; (b) target domain.

**Table 5.** Classification accuracies of the different methods on the CWRU dataset. The numbers in bold indicate the highest classification accuracy in corresponding task.

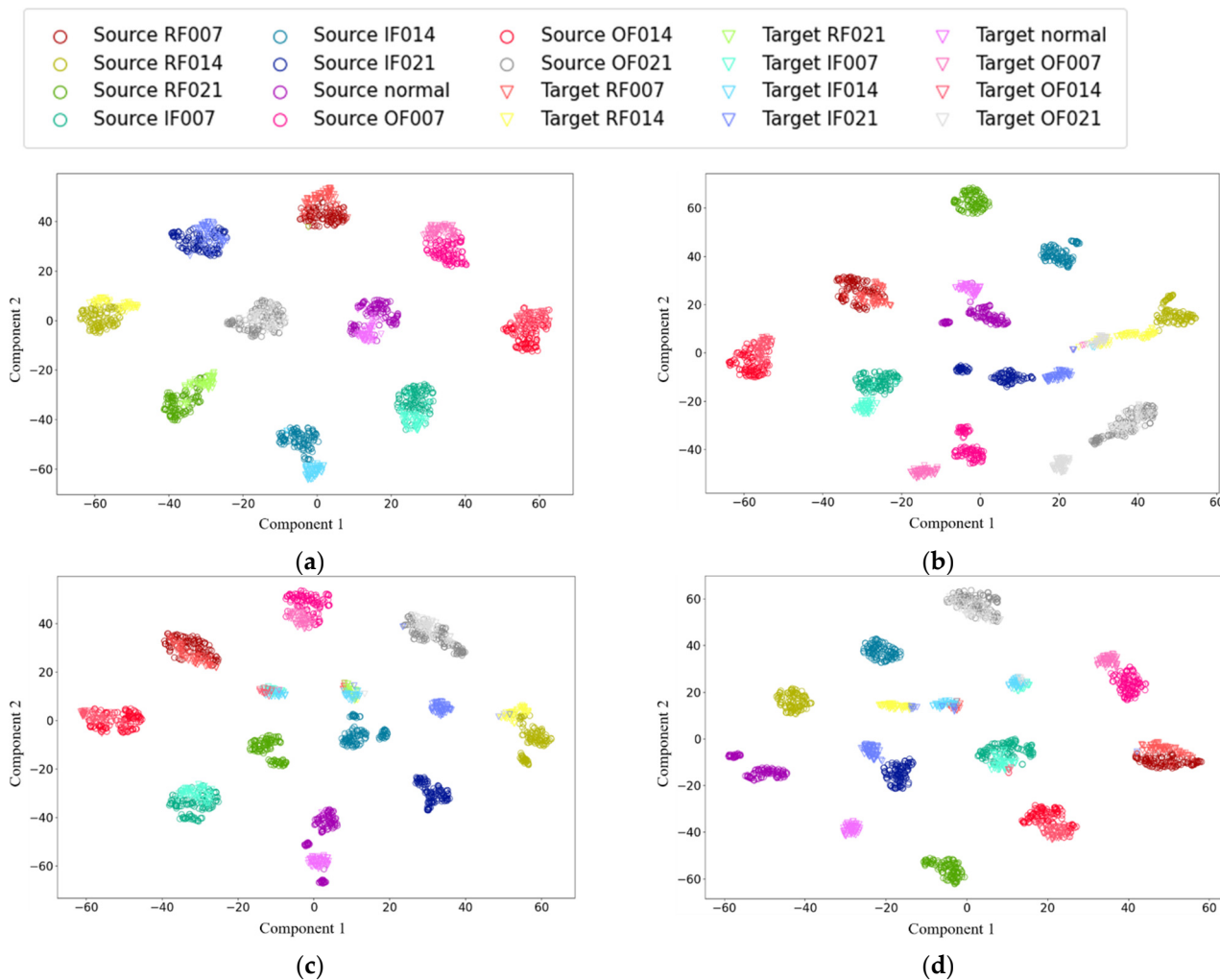
Task	1D-CNN	DDC	DANN	WDMAN [15]	Task	1D-CNN
A-B	99.23%	98.12%	99.53%	99.73%	99.20%	<b>99.90%</b>
A-C	89.20%	93.61%	95.50%	99.67%	99.37%	<b>99.89%</b>
A-D	77.88%	84.36%	84.43%	<b>100.00%</b>	99.37%	99.98%
B-A	98.23%	98.34%	97.39%	99.13%	99.01%	<b>99.96%</b>
B-C	91.59%	98.47%	98.50%	<b>100.00%</b>	99.92%	<b>100.00%</b>
B-D	78.51%	79.41%	88.17%	99.93%	99.31%	<b>99.97%</b>
C-A	88.90%	88.94%	92.70%	98.53%	99.13%	<b>99.77%</b>
C-B	90.66%	92.57%	93.76%	<b>99.80%</b>	99.40%	99.55%
C-D	84.59%	90.03%	90.72%	<b>100.00%</b>	99.40%	99.93%
D-A	77.27%	78.69%	79.19%	98.07%	98.84%	<b>99.54%</b>
D-B	69.82%	72.33%	76.71%	98.27%	99.24%	<b>99.48%</b>
D-C	80.06%	83.61%	86.37%	99.53%	99.61%	<b>99.89%</b>
AVG	85.49%	88.21%	90.25%	99.39%	99.32%	<b>99.82%</b>

Figure 5 shows the confusion matrices of the 1D-LDSAN, 1D-LCNN, DDC, and DANN for the transfer learning task A-D. The proposed method achieves 100% accuracy for each condition, except for the label OF021. As shown in Figure 5b, the 1D-LCNN misclassifies many samples due to the significant distribution discrepancy between the domains. For example, almost all samples of OF014 are misclassified as RF014. In contrast, the DDC and DANN obtain better results than the 1D-LCNN.

**Figure 5.** Confusion matrices of the different models for transfer learning task A–D (the color of the background becomes darker as the values get larger). (a) 1D-LDSAN; (b) 1D-CNN; (c) DDC; (d) DANN.

T-distributed stochastic neighbor embedding (t-SNE) [48] is used for nonlinear dimensionality reduction to visualize the features and analyze the domain adaptation and classification

performance of the models. The visualization results of the 1D-LDSAN, 1D-LCNN, DDC, and DANN for the randomly chosen transfer learning task D-A are shown in Figure 6. It is observed that the proposed 1D-LDSAN produces more separated clusters (Figure 6a) than the 1D-CNN (Figure 6b) (no transfer learning). These results indicate that the 1D-LDSAN can better deal with the domain shift between the source and target domains. In contrast, there are substantial discrepancies between the source and target domains in the two domain adaptation methods (Figure 6c,d), resulting in many misclassifications. In summary, these results demonstrate that the proposed approach achieves more satisfactory classification performance and domain adaptation ability than the other methods.



**Figure 6.** Visualization of features of the different models for the transfer learning task D-A using t-distributed stochastic neighbor embedding. (a) 1D-LDSAN; (b) 1D-CNN; (c) DDC; (d) DANN.

#### 4.4. Verification with a Small Proportion of the Target Domain Data

Different proportions of target domain data used for training produce different results. An experiment was conducted to determine the subdomain adaptation ability of the proposed model using a small proportion of the target domain data. We used six proportions of the target domain data for transfer learning (0%, 10%, 20%, 30%, 40%, and 50%, where 0% indicates no transfer learning). The remaining target domain data were used for testing.

As shown in Table 6, the classification accuracy improved from 0% to 10%, especially for tasks whose working conditions changed substantially. The accuracy of the 1D-LDSAN was high for the last five proportions, indicating that the proposed framework has good generalization performance for different percentages of the target domain data. Furthermore, when only 10% of the unlabeled target domain data were used for training in the

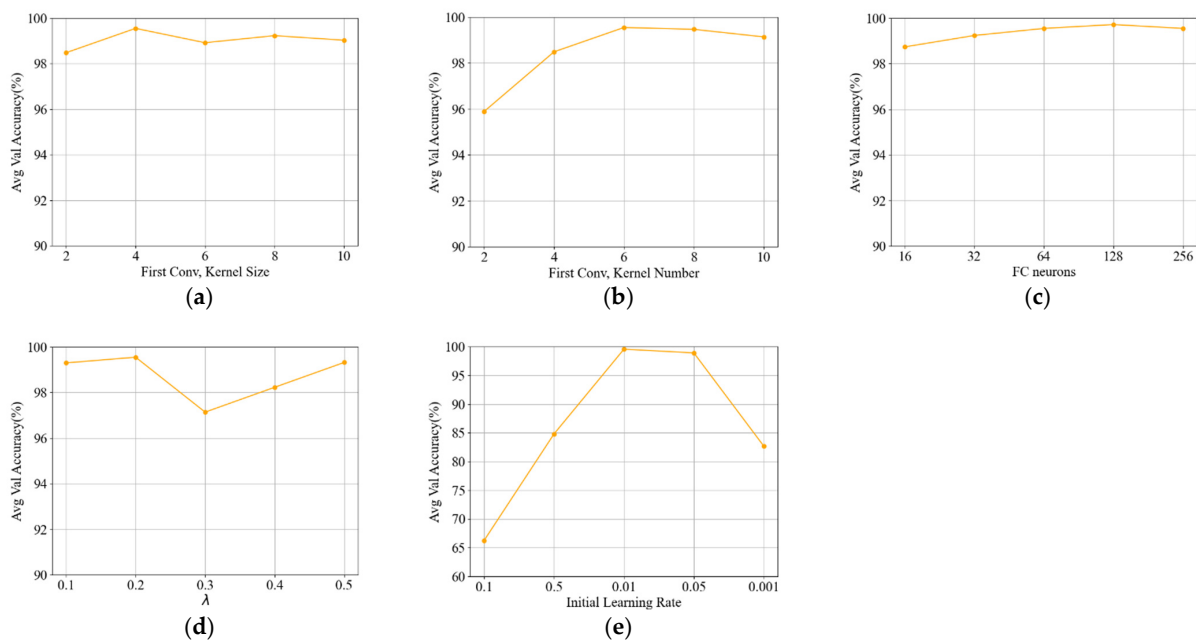
12 transfer tasks, the proposed model achieved more than 98% accuracy based on the remaining 90% of the target domain test data. The experimental results show that the proposed framework has strong feature extraction and domain adaptation ability and can extract sufficient information from a small proportion of the target domain data.

**Table 6.** Experimental results for different proportions of the target domain data. The numbers in bold indicate the highest classification accuracy in corresponding task.

Task	0%	10%	20%	30%	40%	50%
A-B	99.23%	99.79%	99.57%	99.84%	<b>99.92%</b>	99.90%
A-C	89.20%	99.90%	99.79%	99.90%	<b>99.94%</b>	99.89%
A-D	77.88%	98.79%	98.85%	99.58%	97.48%	<b>99.98%</b>
B-A	98.23%	99.90%	99.86%	99.76%	99.91%	<b>99.96%</b>
B-C	91.59%	99.78%	99.96%	99.98%	99.75%	<b>100.00%</b>
B-D	78.51%	99.94%	99.27%	99.10%	<b>99.99%</b>	99.97%
C-A	88.90%	99.47%	99.70%	99.71%	<b>99.87%</b>	99.77%
C-B	90.66%	99.52%	99.53%	99.57%	<b>99.65%</b>	99.55%
C-D	84.59%	99.48%	99.66%	99.78%	99.82%	<b>99.93%</b>
D-A	77.27%	99.20%	<b>99.61%</b>	98.30%	99.33%	99.54%
D-B	69.82%	98.09%	98.38%	99.30%	99.09%	<b>99.48%</b>
D-C	80.06%	99.68%	99.78%	99.53%	99.79%	<b>99.89%</b>
AVG	85.50%	99.46%	99.49%	99.53%	99.55%	<b>99.82%</b>

#### 4.5. Parameter Sensitivity Analysis

A sensitivity analysis of the five key parameters of the proposed framework was conducted. The results for the validation task are presented in Figure 7. The influence of multiple parameters was examined. It was found that the detailed framework architecture had a negligible influence on the model performance, except for the kernel number of the first convolution layer. The effects of the threshold parameter  $\lambda$  and the initial learning rate were also investigated. The results indicate that  $\lambda$  has a relatively small influence on the framework performance. It is worth noting that the initial learning rate has a marked influence on model performance.



**Figure 7.** Effects of different parameters on the framework performance. (a) Kernel size of the first convolution layer; (b) kernel number of the first convolution layer; (c) neurons of the FC layer; (d)  $\lambda$ ; (e) initial learning rate.

## 5. Conclusions

We proposed the one-dimensional lightweight deep subdomain adaptation network (1D-LDSAN) to classify fault types of rolling bearings under different working conditions. The raw vibration signal was divided into small segments in the source and target domains. The advanced features in the segments were extracted by a one-dimensional lightweight convolutional neural network backbone. The local maximum mean discrepancy (LMMD) was employed to match the subdomain distributions, and the cross-entropy function was used to train a fully connected classifier using the labeled source domain data.

We compared the classification accuracy for different signal lengths and chose a length of 1024. The proposed 1D-LDSAN framework outperformed five other models for classifying rolling bearing faults on the CWRU dataset, indicating superior diagnosis performance. An experiment with six proportions of the target domain data for training indicated that the proposed framework could extract sufficient information from a small proportion of the target domain data, indicating excellent domain adaptation performance.

This study provides a solution for the intelligent fault diagnosis of rolling bearings and demonstrates the potential of domain adaptation for fault diagnosis under different working conditions. In a future study, we will focus on more effective deep domain adaptation methods.

**Author Contributions:** Conceptualization: Y.G.; Methodology: Y.G.; Software: R.Z.; Validation: R.Z.; Formal Analysis: R.Z. and Y.G.; Investigation: Y.G.; Resources: Y.G.; Data Curation: Y.G.; Writing—Original Draft: R.Z. and Y.G.; Writing—Review and Editing: R.Z. and Y.G.; Visualization: Y.G.; Supervision: Y.G.; Project Administration: Y.G.; Funding Acquisition: Y.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** The authors would like to thank the National Natural Science Foundation of China (Grant No. 61876059).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

## References

1. Tang, S.; Yuan, S.; Zhu, Y. Convolutional Neural Network in Intelligent Fault Diagnosis toward Rotatory Machinery. *IEEE Access* **2020**, *8*, 86510–86519. [\[CrossRef\]](#)
2. Lang, X.; Steven, C.; Paolo, P. Rolling element bearing diagnosis based on singular value decomposition and composite squared envelope spectrum. *Mech. Syst. Signal Process.* **2021**, *148*, 107174.
3. Hoang, D.T.; Kang, H.J. A survey on Deep Learning based bearing fault diagnosis. *Neurocomputing* **2019**, *335*, 327–335. [\[CrossRef\]](#)
4. Cerrada, M.; Sanchez, R.V.; Li, C.; Pacheco, F.; Cabrera, D.; Oliveira, J.V.D.; Vasquez, R. A review on data-driven fault severity assessment in rolling bearings. *Mech. Syst. Signal Process.* **2018**, *99*, 169–196. [\[CrossRef\]](#)
5. Zhang, S.; Zhang, S.; Wang, B.; Thomas, G.H. Deep Learning Algorithms for Bearing Fault Diagnostics—A Comprehensive Review. *IEEE Access* **2020**, *8*, 29857–29881. [\[CrossRef\]](#)
6. Xu, X.; Cao, D.; Zhou, Y.; Gao, J. Application of neural network algorithm in fault diagnosis of mechanical intelligence. *Mech. Syst. Signal Process.* **2020**, *141*, 106625. [\[CrossRef\]](#)
7. Testa, A.; Cinque, M.; Coronato, A.; Pietro, G.D.; Augusto, J.C. Heuristic strategies for assessing wireless sensor network resiliency: An event-based formal approach. *J. Heuristics* **2015**, *21*, 145–175. [\[CrossRef\]](#)
8. Wei, H.; Zhang, Q.; Shang, M.; Gu, Y. Extreme learning Machine-based classifier for fault diagnosis of rotating Machinery using a residual network and continuous wavelet transform. *Measurement* **2021**, *183*, 109864. [\[CrossRef\]](#)
9. 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. [\[CrossRef\]](#)
10. Neupane, D.; Seok, J. Bearing Fault Detection and Diagnosis Using Case Western Reserve University Dataset with Deep Learning Approaches: A Review. *IEEE Access* **2020**, *8*, 93155–93178. [\[CrossRef\]](#)
11. Song, X.; Zhu, D.; Liang, P.; An, L. A New Bearing Fault Diagnosis Method Using Elastic Net Transfer Learning and LSTM. *J. Intell. Fuzzy Syst.* **2021**, *40*, 12361–12369. [\[CrossRef\]](#)



12. Shen, C.; Xie, J.; Wang, D.; Jiang, X.; Shi, J.; Zhu, Z. Improved Hierarchical Adaptive Deep Belief Network for Bearing Fault Diagnosis. *Appl. Sci.* **2019**, *9*, 3374. [[CrossRef](#)]
13. Shao, S.; Yan, R.; Lu, Y.; Wang, P.; Gao, R.X. DCNN-Based Multi-Signal Induction Motor Fault Diagnosis. *IEEE Trans. Instrum. Meas.* **2020**, *6*, 2658–2669. [[CrossRef](#)]
14. Jv, A.; Yqc, A.; Jing, W.B. FaultFace: Deep Convolutional Generative Adversarial Network (DCGAN) based Ball-Bearing failure detection method. *Inf. Sci.* **2021**, *542*, 195–211.
15. Ince, T.; Kiranyaz, S.; Eren, L.; Askar, M.; Gabbouj, M. Real-Time Motor Fault Detection by 1-D Convolutional Neural Networks. *IEEE Trans. Ind. Electron.* **2016**, *63*, 7067–7075. [[CrossRef](#)]
16. Cheng, C.; Zhou, B.; Ma, G.; Wu, D.; Yuan, Y. Wasserstein distance based deep adversarial transfer learning for intelligent fault diagnosis with unlabeled or insufficient labeled data. *Neurocomputing* **2020**, *409*, 35–45. [[CrossRef](#)]
17. Zhang, B.; Li, W.; Li, X.; Ng, S. Intelligent Fault Diagnosis Under Varying Working Conditions Based on Domain Adaptive Convolutional Neural Networks. *IEEE Access* **2018**, *6*, 66367–66384. [[CrossRef](#)]
18. Wang, K.; Wei, Z.; Xu, A.; Zeng, P.; Yang, S. One-Dimensional Multi-Scale Domain Adaptive Network for Bearing-Fault Diagnosis under Varying Working Conditions. *Sensors* **2020**, *21*, 6039. [[CrossRef](#)]
19. Pan, S.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [[CrossRef](#)]
20. Ganin, Y.; Lempitsky, V. Unsupervised domain adaptation by backpropagation. *arXiv* **2014**, arXiv:1409.7495.
21. Yang, B.; Li, Q.; Chen, L.; Shen, C. Bearing Fault Diagnosis Based on Multilayer Domain Adaptation. *Shock Vib.* **2020**, *2020*, 8873960. [[CrossRef](#)]
22. Wu, J.; Tang, T.; Chen, M.; Wang, Y.; Wang, K. A study on adaptation lightweight architecture based deep learning models for bearing fault diagnosis under varying working conditions. *Expert Syst. Appl.* **2020**, *160*, 113710. [[CrossRef](#)]
23. Zhang, W.; Li, X.; Ma, H.; Luo, Z.; Li, X. Open-Set Domain Adaptation in Machinery Fault Diagnostics Using Instance-Level Weighted Adversarial Learning. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7445–7455. [[CrossRef](#)]
24. Jiao, J.; Zhao, M.; Lin, J.; Liang, K. Residual joint adaptation adversarial network for intelligent transfer fault diagnosis. *Mech. Syst. Signal Process.* **2020**, *145*, 106962. [[CrossRef](#)]
25. Zhu, Y.; Zhuang, F.; Wang, J.; Ke, G.; Chen, J.; Bian, J.; Xiong, H.; He, Q. Deep Subdomain Adaptation Network for Image Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 1713–1722. [[CrossRef](#)] [[PubMed](#)]
26. Zhang, W.; Peng, G.; Li, C.; Chen, Y.; Zhang, Z. A New Deep Learning Model for Fault Diagnosis with Good Anti-Noise and Domain Adaptation Ability on Raw Vibration Signals. *Sensors* **2017**, *2*, 425. [[CrossRef](#)] [[PubMed](#)]
27. Yu, D.; Gu, Y. A Machine Learning Method for the Fine-Grained Classification of Green Tea with Geographical Indication Using a MOS-Based Electronic Nose. *Foods* **2021**, *10*, 795. [[CrossRef](#)]
28. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 37, pp. 448–456.
29. He, Z.; Shao, H.; Zhong, X.; Zhao, X. Ensemble transfer CNNs driven by multi-channel signals for fault diagnosis of rotating machinery cross working conditions. *Knowl.-Based Syst.* **2020**, *207*, 106396. [[CrossRef](#)]
30. Glorot, X.; Antoine, B.; Yoshua, B. Deep Sparse Rectifier Neural Networks. *J. Mach. Learn. Res.* **2011**, *15*, 315–323.
31. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2017**, *60*, 84–90. [[CrossRef](#)]
32. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
33. LeCun, Y.; Boser, B.; Denker, J.; Henderson, D.; Howard, R.; Hubbard, W.; Jackel, L. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [[CrossRef](#)]
34. Yu, C.; Wang, J.; Chen, Y.; Huang, M. Transfer Learning with Dynamic Adversarial Adaptation Network. In Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM), Beijing, China, 8–11 November 2019; pp. 778–786.
35. Long, M.; Cao, Y.; Wang, J.; Jordan, M.I. Learning transferable features with deep adaptation networks. In Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML'15), Lille, France, 6–11 July 2015; pp. 97–105.
36. 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. Inform.* **2021**, *17*, 7957–7967. [[CrossRef](#)]
37. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, CA, USA, 26 June–1 July 2016; pp. 770–778.
39. Zheng, H.; Gu, Y. EnCNN-UPMWS: Waste Classification by a CNN Ensemble Using the UPM Weighting Strategy. *Electronics* **2021**, *10*, 427. [[CrossRef](#)]
40. Pan, S.J.; Tsang, I.W.; Kwok, J.T.; Yang, Q. Domain Adaptation via Transfer Component Analysis. *IEEE Trans. Neural Netw.* **2011**, *22*, 199–210. [[CrossRef](#)]
41. Boer, P.; Kroese, D.P.; Mannor, S.; Rubinstein, R.A. Tutorial on the Cross-Entropy Method. *Ann. Oper. Res.* **2005**, *134*, 19–67. [[CrossRef](#)]

42. Li, Q.; Shen, C.; Chen, L.; Zhu, Z. Knowledge mapping-based adversarial domain adaptation: A novel fault diagnosis method with high generalizability under variable working conditions. *Mech. Syst. Signal Process.* **2021**, *147*, 107095. [[CrossRef](#)]
43. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
44. Case Western Reserve University Bearing Dataset. Available online: <https://engineering.case.edu/bearingdatacenter> (accessed on 21 January 2022).
45. Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep Domain Confusion: Maximizing for Domain Invariance. *arXiv* **2014**, arXiv:1412.3474.
46. 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*, 1–35.
47. 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](#)]
48. Laurens, V.D.M.; Hinton, G. Visualizing data using t-SNE. *Mach. Learn. Res.* **2008**, *9*, 2579–2605.