

## Article

# A Multi-Adversarial Joint Distribution Adaptation Method for Bearing Fault Diagnosis under Variable Working Conditions

Zhichao Cui <sup>1</sup>, Hui Cao <sup>1,2,\*</sup>, Zeren Ai <sup>1</sup> and Jihui Wang <sup>1</sup>

<sup>1</sup> Marine Engineering College, Dalian Maritime University, Dalian 116026, China; czc1120210080@dlnu.edu.cn (Z.C.); aizeren666@gmail.com (Z.A.); wjh921130@163.com (J.W.)

<sup>2</sup> Dalian Maritime University Smart Ship Limited Company, Dalian 116026, China

\* Correspondence: lustersoft@foxmail.com

**Abstract:** Deep network fault diagnosis requires a lot of labeled data and assumes identical data distributions for training and testing. In industry, varying equipment conditions lead to different data distributions, making it challenging to maintain consistent fault diagnosis performance across conditions. To this end, this paper designs a transfer learning model named the multi-adversarial joint distribution adaptation network (MAJDAN) to achieve effective fault diagnosis across operating conditions. MAJDAN uses a one-dimensional lightweight convolutional neural network (1DLCNN) to directly extract features from the original bearing vibration signal. Combining the distance-based domain-adaptive method, maximum mean difference (MMD), with the multi-adversarial network will simultaneously reduce the conditional and marginal distribution differences between the domains. As a result, MAJDAN can efficiently acquire domain-invariant feature information, addressing the challenge of cross-domain bearing fault diagnosis. The effectiveness of the model was verified based on two sets of different bearing vibration signals, and one-to-one and one-to-many working condition migration task experiments were carried out. Simultaneously, various levels of noise were introduced to the signal to enable analysis and comparison. The findings demonstrate that the suggested approach achieves exceptional diagnostic accuracy and exhibits robustness.

**Keywords:** intelligent diagnosis; multi-adversarial domain adaptation; deep transfer learning; rolling bearing



**Citation:** Cui, Z.; Cao, H.; Ai, Z.; Wang, J. A Multi-Adversarial Joint Distribution Adaptation Method for Bearing Fault Diagnosis under Variable Working Conditions. *Appl. Sci.* **2023**, *13*, 10606. <https://doi.org/10.3390/app131910606>

Academic Editor: Jose Machado

Received: 15 August 2023

Revised: 8 September 2023

Accepted: 21 September 2023

Published: 23 September 2023



**Copyright:** © 2023 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/>).

## 1. Introduction

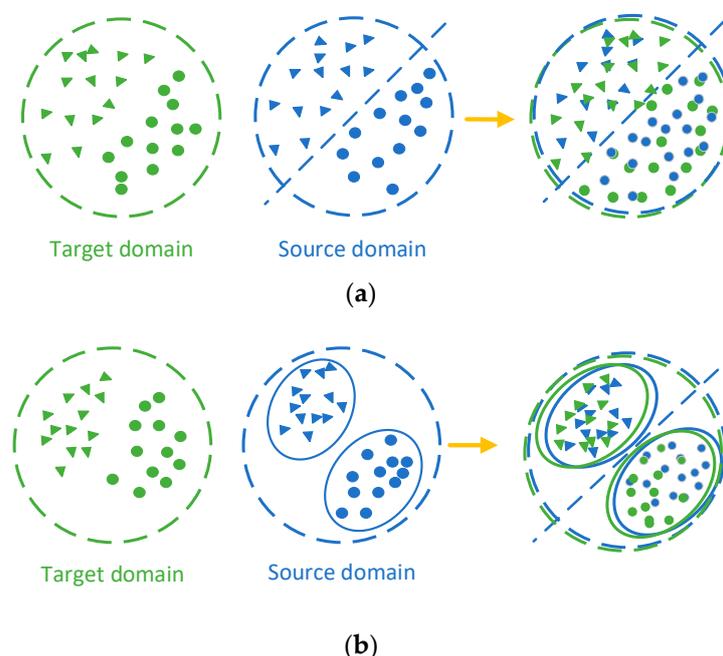
Bearings play a vital role in the functioning of rotating machinery to minimize friction between mechanical components. The detrimental effects of bearing damage are especially pronounced in various rotating machinery, often resulting in significant mechanical failures [1]. Therefore, it is very important to ensure the safety and stability of bearings through health monitoring and intelligent fault diagnosis [2–4]. Researchers have conducted extensive studies on fault diagnosis methods, which include methods based on signal processing, statistical analysis, and machine learning. Methods based on signal processing utilize time–frequency analysis to extract fault characteristic information from vibration signals for fault diagnosis. Common techniques include Short-Time Fourier Transform [5], wavelet transform [6], and mathematical morphology analysis [7], etc. Methods based on statistical analysis monitor and analyze the characteristic values of signals, extracting fault characteristic quantities from the changes produced and comparing them with designed standard values and thresholds to determine the faults. The primary methods include Bayesian theory [8] and multivariate statistical analysis, etc. The method based on machine learning trains the model by collecting normal operation data and fault data so that the model can automatically learn fault characteristics. Compared to the previous two methods, machine learning does not rely heavily on expert knowledge and experience. In fault diagnosis applications, it offers greater flexibility and universality, as well as higher efficiency and accuracy.

With the advancement of machine learning algorithms, techniques for bearing fault diagnosis have also made significant progress. Among these techniques, deep learning-based diagnosis methods have gained popularity in recent years, demonstrating superior performance compared to shallow learning methods [9]. One of the main reasons is that deep learning eliminates the part of manual feature extraction, automatically learns internal representation and predict targets from the original input, and realizes an end-to-end system [10]. Numerous deep learning models have been employed in the realm of fault diagnosis, with convolutional neural networks (CNN), recurrent neural networks (RNN), and their respective variants being particularly prominent [11–13]. Ding et al. introduced the transformer network based on the self-attention mechanism into bearing fault diagnosis, achieving satisfactory diagnostic efficiency and results [14]. Zhou et al. integrated generative adversarial networks with deep convolution, utilizing a small amount of labeled data and a large amount of unlabeled data for semi-supervised learning, and attempted to diagnose on unlabeled data with expanded fault types [15].

Typically, fault diagnosis models assume that the training data and the test data follow the same distribution [16,17]. As a result, the trained models generally achieve high accuracy when evaluated on the test dataset. However, in the actual industry, the working conditions and states of mechanical equipment are not consistent, which will lead to changes in the data distribution under different working states [18]. When a model trained on a dataset from one specific working condition is utilized for diagnosing data from different working conditions, the diagnosis results often prove unsatisfactory, even when the fault type remains the same [19]. For this issue, Zhou et al. introduced a probabilistic Bayesian deep network to quantify and analyze the uncertainty of the model's diagnostic results under unknown conditions. The results indicate that the model has a high level of uncertainty in predicting failures under unknown conditions, and the larger the difference between conditions, the greater the uncertainty [20]. In addition, the training of the deep model mainly relies on the data with fault labels to adjust the weight of the model and then find out the mapping function that corresponds the data to the label. But labeling data consumes a lot of labor costs. To tackle this problem and achieve a solution, the concept of transfer learning was developed. The key advantage of transfer learning lies in its ability to leverage labeled data (source domain) from one specific working condition and unlabeled data (target domain) from other, related working conditions for training purposes [21]. It is worth noting that this differs from semi-supervised learning, which only involves a single working condition. Semi-supervised learning trains models by leveraging a combination of a small amount of labeled data and a large volume of unlabeled data, addressing the issue of limited labeled data. In contrast, transfer learning involves source and target domain data from different working conditions. The crux of transfer learning lies in addressing the distributional discrepancy between the two domains. This approach empowers the model to be utilized for fault diagnosis across diverse working conditions. Therefore, transfer learning can build bridges between different fields that follow different probability distributions and establish a learning mechanism that spans different fields [22].

Within the domain of transfer learning, unsupervised domain adaptation (UDA) holds significant importance and has been extensively explored by researchers. It has found successful applications in various fields, including computer vision [23], target detection [24], etc. Since UDA has shown promising results in transfer learning, many researchers have introduced UDA into the field of mechanical fault diagnosis to solve the problem of fault diagnosis under different operating states. UDA can be broadly categorized into two types: base distance and base adversarial [25]. Distance-based unsupervised domain adaptation (UDA) involves mapping features into a shared feature space and employing methods such as Correlation Alignment (CORAL) [26], maximum mean discrepancy (MMD) [27], Central Moment Discrepancy (CMD) [28], and other difference measurement techniques. These methods estimate the distribution gap between the source domain and the target domain, thereby minimizing the distribution discrepancy. The adversarial-based UDA utilizes the interaction between the feature extraction module and the domain discrimination module

to achieve the purpose of domain confusion, thereby extracting invariant features from two domains [29]. Inspired by the above methods, many different pieces of research have designed a UDA algorithm for cross-operating condition fault diagnosis. Li et al. employed CNN for extracting features from rolling bearings. They then minimized the MMD between the multi-layer models and effectively narrowed the distribution distance of feature mappings between the source domain and the target domain [30]. Yu et al. utilized a domain adversarial method based on the Wasserstein distance to train raw signal data, reducing the marginal distribution between the source and target domains in the high-dimensional feature space to extract domain-invariant features [31]. Qin et al. introduced a combination of confrontation and CORAL techniques to tackle the challenge of cross-domain fault diagnosis in gearboxes [32]. However, most of the existing transfer learning methods align marginal distributions (Figure 1a) or conditional distributions (Figure 1b) independently, and they rarely consider both at the same time.



**Figure 1.** (a) Marginal distribution alignment, and (b) conditional distribution alignment.

To tackle the previously mentioned challenges and improve the effectiveness of cross-domain fault diagnosis, this article propose an innovative approach based on the multi-adversarial joint distribution adaptation network (MAJDAN). The main contributions of this article are as follows:

- (1) Through the integration of maximum mean discrepancy (MMD) and multi-adversarial networks for domain adaptation, our model can concurrently align both conditional and marginal distributions. The advantage of this combined approach is that it not only leverages the capabilities of adversarial learning to capture complex non-linear feature mappings but also utilizes the statistical properties of MMD to ensure that the distributions of the source and target domains are close in the feature space. This can offer a more robust and resilient domain adaptation method.
- (2) We employed a 1D lightweight CNN as a feature extractor to directly learn from raw loyalty signals, offering a computational advantage over traditional deep convolutional neural networks.
- (3) We validated our proposed model on two datasets across different transfer tasks, confirming its effectiveness. We also introduced noise to vibration signals to assess the model's resilience against noise, and comparisons were made with other approaches.

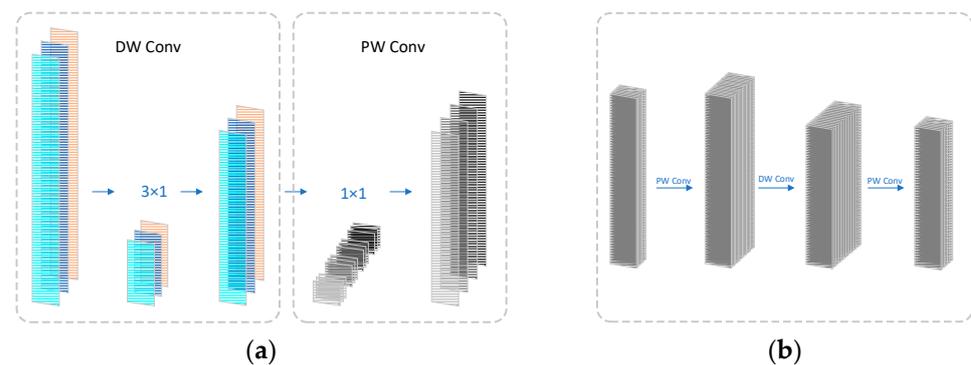
The structure of the remainder of this paper is as follows: In Section 2, a detailed exposition of the theories used in the proposed method is provided. Section 3 elucidates

the structure and functionality of each module within MAJDAN and presents the final loss function, detailing the algorithm's training process. Section 4 offers the experimental setup and results analysis of the algorithm on the dataset. Section 5 concludes the paper.

## 2. Preliminary Knowledge

### 2.1. One-Dimensional Lightweight Convolution

Convolutional neural networks (CNNs) are mostly used to solve image problems, so many fault diagnosis methods use 2D CNNs as a feature extraction method. Many researchers use various techniques to convert one-dimensional signals received by sensors into two-dimensional images for direct input into existing 1D CNNs for feature extraction. For example, convert one-dimensional sequence data into two-dimensional grayscale images [33,34]; use continuous wavelet transform or fast Fourier transform methods to convert one-dimensional sequence data into time–frequency images [35,36]. These conversion methods have achieved satisfactory results in fault diagnosis, but converting 1D sequence data to 2D images consumes additional computational cost, and the training cost of 2D CNNs is also higher than that of 1D CNNs. Therefore, this paper uses a lightweight 1D CNN to directly extract features from the original data. 1DLCNN refers to the lightweight image classification network MobileNet V2 [37], which can reduce computing costs while ensuring accuracy [38]. Figure 2 shows two special convolutional blocks contained in lightweight convolutional neural networks: the depthwise separable convolution block and the inverted residual block. The fundamental ideas and tenets of each module in the lightweight CNNs are explained briefly in this section.



**Figure 2.** Convolutional blocks in lightweight convolutional neural networks. (a) Depthwise separable convolution; (b) inverted residual block.

The input of a standard 1D CNN is a vector and a convolution kernel, where the convolution kernel's size is often smaller than the input, and this configuration creates a local receptive field. Weight sharing refers to the fact that the filter's weights remain constant when each filter moves over the input map. However, the number of parameters grows with deeper convolutional layers, increasing the computing cost. Lightweight CNNs are based on depthwise separable convolutions, which are a form of factorized convolutions [39]. This form decomposes the standard convolution into a depthwise convolution (DW), which operates channel-wise, and a pointwise convolution (PW) with a  $1 \times 1$  kernel. DW convolution differs from standard convolution in that it employs a single-channel convolution kernel. The feature maps generated through DW convolution have the same number of channels as the input. Then, the feature maps after DW convolution are combined with PW convolution to generate new feature maps. Figure 1 illustrates the computational process of depthwise separable convolution, which divides the conventional convolution operation into two sequential steps: filtering and combining. This approach

effectively reduces the computational burden. Comparing this to a standard convolution can be expressed as follows:

$$\frac{C_I \cdot K_1 \cdot K_2 \cdot F_1 \cdot F_2 + C_I \cdot C_O \cdot F_1 \cdot F_2}{C_I \cdot C_O \cdot K_1 \cdot K_2 \cdot F_1 \cdot F_2} = \frac{1}{C_O} + \frac{1}{K_1 \cdot K_2} \tag{1}$$

where  $C_I$  denotes the count of input channels,  $C_O$  represents the number of output channels,  $K_1 \times K_2$  is the size of the convolution kernel, and  $F_1 \times F_2$  is the size of the feature map. The numerator is the computational cost of depthwise separable convolution, and the denominator is the computational cost of standard convolution. It can be seen from the simplified results that the depthwise separable convolution can save a lot of computational costs compared with the traditional convolution, and it also shows a good feature extraction ability in the model of this paper.

The inverted residual structure can be seen as the inverse or opposite of the residual structure, and the  $1 \times 1$  convolution kernel is employed to expand the dimension. Following that, DW convolution extracts features, and ultimately, the  $1 \times 1$  convolution reduces the dimension. The operation of increasing dimensionality is performed to extract more information. Due to the limitation of DW convolution, which does not alter the number of channels, a dimensionality expansion is initially performed to increase the channel count. Subsequently, DW convolution is applied in the expanded high-dimensional space to extract additional feature information.

### 2.2. Multi-Adversarial Adaptation

In the realm of cross-condition mechanical fault diagnosis, the domain adaptation techniques that utilizes adversarial networks to perform domain confusion by extracting domain-invariant features has achieved commendable results. Domain adaptation is defined formally as the following: given a source domain  $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$  of  $n_s$  labeled samples and a target domain  $D_t = \{x_j^t\}_{j=1}^{n_t}$  of  $n_t$  unlabeled samples. The data from both domains are sampled from joint distributions  $P(X^s, Y^s)$  and  $Q(X^t, Y^t)$ , respectively, where  $P \neq Q$ . Utilize the data from the source domain to train a predictive function  $f$  for the target domain, represented as  $f : x_t \rightarrow y_t$ , so that  $f$  has the lowest prediction error in the target domain.

The adversarial learning process can be conceptualized as a two-player game. In this game, the two players are the domain discriminator,  $G_d$ , and the feature extractor,  $G_f$ . While  $G_d$  strives to identify the domain of each sample,  $G_f$  aims to extract shared features across both domains, constantly challenging the capabilities of the domain discriminator. For the adversarial learning objective function, the aim is to minimize the classification loss associated with  $G_f$  and simultaneously maximize the loss for  $G_d$ . This process ensures that the parameters of the feature extractor are adjusted specifically to derive domain-invariant features.

However, a single domain classifier can only align the distributions of the two domains on a global scale. After global feature alignment, it is indeed possible to achieve decent classification results for two domains with similar overall distributions. But merely a global alignment of domains lacks fine-grained information and makes it difficult to distinguish between the structural properties of each category, which potentially leads to a misclassification of samples near the category decision boundary [40].

Multi-adversarial domain adaption has  $K$  domain discriminators. Each domain discriminator can be called a local domain discriminator  $G_d^k$  ( $k = 1, \dots, K$ ).  $K$  corresponds to the class of the sample, and the  $K$ th domain discriminator is responsible for distinguishing the domains of the samples associated with the  $K$ th class, as shown in Figure 3. Although the target domain data are unlabeled, the label classifier outputs a predicted label  $\hat{y}_i = G_y(x_i)$ .  $\hat{y}_i$  is actually the probability of sample  $x_i$  over all classes. This probability guides the significance of each sample  $x_i$  to the specific local domain discriminator  $G_d^k$ . The weight of the feature  $G_f(x_i)$  in relation to the local domain discriminator  $G_d^k$  is determined by the

probability  $\hat{y}_i$ , reflecting the importance of each sample. The mathematical expression to apply it to all local discriminators is as follows:

$$L_d = \frac{1}{n} \sum_{k=1}^K \sum_{x_i \in \mathcal{D}_s \cup \mathcal{D}_t} L_d^k \left( G_d^k \left( \hat{y}_i^k G_f(x_i) \right), d_i \right) \tag{2}$$

where  $G_d^k$  is the  $k$ th local domain discriminator,  $L_d^k$  is its cross-entropy loss, and  $i$  is the domain label of  $x_i$ ,  $n = n_s + n_t$ .

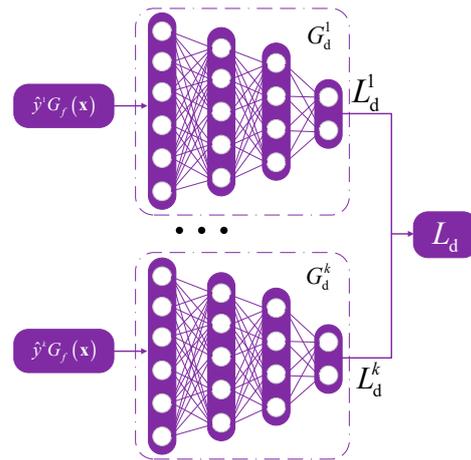


Figure 3. Structure of domain discriminator.

### 2.3. Maximum Mean Difference (MMD)

Domain adaptation methods based on discrepancy metrics play an important role in transfer learning, where the most widely used discrepancy metric is the maximum mean difference (MMD). Many domain adaptation methods employ the MMD to quantify the distributional divergence between source and target domains, and for further improvement on this basis, a plethora of UDA methods have been developed.

MMD is utilized to quantify the dissimilarity between two distinct yet related distributions. Given that the data from the source and target domains originate from two different distributions,  $p$  and  $q$ , respectively, MMD calculates the distance between these distributions within the Reproducing Kernel Hilbert Space (RKHS) [41]. Formally, the MMD is defined as follows:

$$L_{MMD}(p, q) \triangleq \left\| E_p[\phi(x^s)] - E_q[\phi(x^t)] \right\|_{\mathcal{H}}^2 \tag{3}$$

where  $\phi(\cdot)$  is the mapping function from the input features to the RKHS. At this time, the sample features of the source domain and the target domain satisfy the equation  $k(x^s, x^t) = \phi(x^s) \cdot \phi(x^t)$ ,  $k(x^s, x^t)$  is the kernel function. Theoretically,  $L_{MMD}(p, q) = 0$  when  $p$  and  $q$  have the same distribution [41].

$L_{MMD}(p, q)$  can be calculated as follows:

$$\begin{aligned} L_{MMD}(p, q) &= \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(x_i) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(x_j) \right\|_{\mathcal{H}}^2 \\ &= \frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} k(x_i^s, x_j^s) \\ &\quad + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} k(x_i^t, x_j^t) \\ &\quad - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} k(x_i^s, x_j^t) \end{aligned} \tag{4}$$

In domain adaptation, the feature extractor’s parameters are adjusted by reducing the MMD between the source and target domain feature representations. The objective of this process is to make the feature representations of the source domain and target domain close to each other, so that when dealing with data from different domains, the feature extractor can generate universal feature representations, which are also known as domain-invariant features. In this paper, MMD is employed to bridge the marginal distribution gap between the source and target domains, ensuring a global feature alignment.

### 3. Multi-Adversarial Joint Distribution Adaptation Network

Three components make up the MAJDAN model: a shared feature extractor  $G_f$  with the parameter  $\theta_f$ , a shared label predictor  $G_c$  with the parameter  $\theta_c$ , and a domain discriminator  $G_d$  with the parameter  $\theta_d$ . Its structure is shown in Figure 4. A 1DLCNN is designed as a feature extractor that directly processes the original signal of bearings and extracts the feature representation. Then, the extracted features are fed into a label predictor and a local domain classifier. In the label predictor, the supervised classification training is performed on the labeled source domain data, and the predicted pseudo-label is output for the unlabeled target domain data. At the same time, MMD is introduced to quantify the global distribution shift between the two domains. A local domain classifier performs subdomain obfuscation for each category based on pseudo-labels.

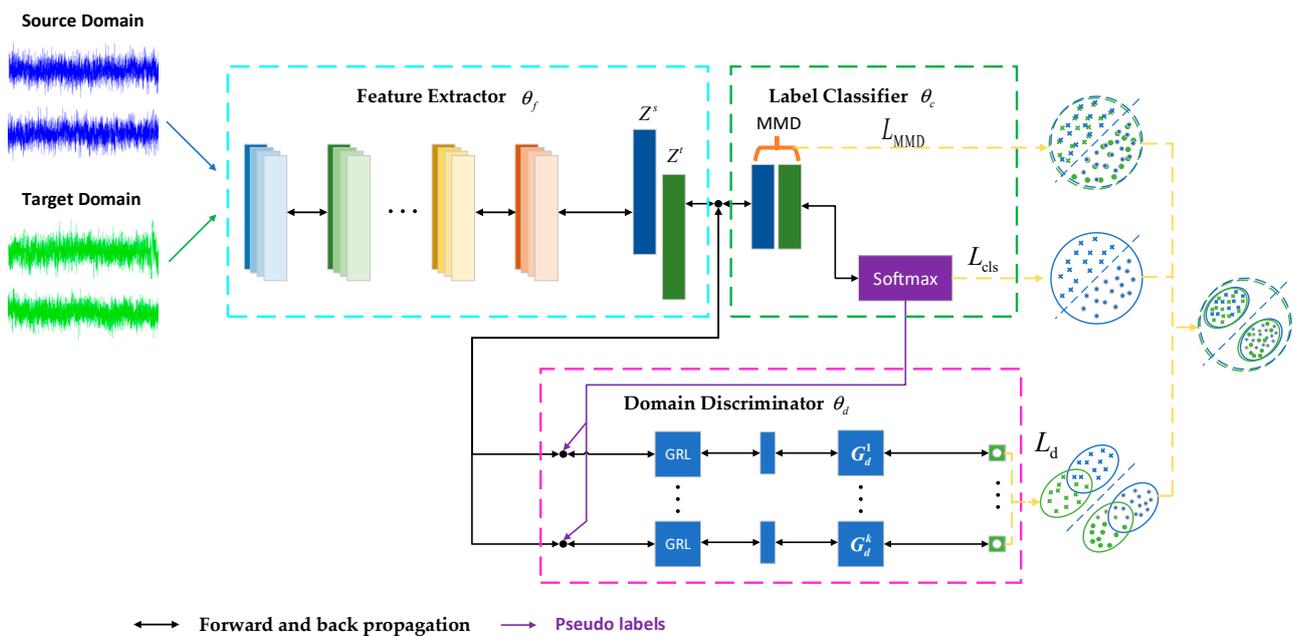


Figure 4. MAJDAN model structure.

#### 3.1. Feature Extractor

The feature extractor of MAJDAN is a 1DLCNN design based on MobileNet V2, which saves computational costs compared with the standard 1D CNN, and also has good feature extraction capabilities. In the first layer, standard convolution is applied to process the input data. Considering the length of the data, a  $46 \times 1$  wide convolutional kernel was chosen to give the model a larger receptive field, capturing features over a broader range and simultaneously reducing the data’s length. In subsequent structures, depthwise separable convolution replaces standard convolution to reduce computational costs. The parameters of the feature extractor were determined through empirical values and manual tuning. At the end of the network, global average pooling is used to reduce the dimensionality of the feature maps. Table 1 shows the parameters of 1DLCNN.

**Table 1.** Parameters of 1DLCNN.

Layer	Convolution Kernel Parameters (n × h × c)	Stride
ConvBNReLU6	6 × 46 × 1	4
ConvBNReLU6	6 × 3 × 1	1
ConvBN	16 × 1 × 6	1
ConvBNReLU6	96 × 1 × 16	1
ConvBNReLU6	96 × 3 × 1	2
ConvBN	24 × 1 × 96	1
ConvBNReLU6	144 × 1 × 24	1
ConvBNReLU6	144 × 3 × 1	2
ConvBN	32 × 1 × 144	1
ConvBNReLU6	32 × 3 × 1	1
ConvBN	48 × 1 × 32	1
ConvBNReLU6	64 × 1 × 48	1
Global Average Pooling	/	/

In our model, the input of the feature extractor  $G_f$  is the one-dimensional vibration signal of the source domain and the target domain. During the training phase, data from the two domains are fed into the model in mini-batches. The inputs from the source and target domains are represented as  $x^s$  and  $x^t$ , respectively. Their corresponding output features are denoted as  $F^s = G_f(x^s; \theta_f)$  and  $F^t = G_f(x^t; \theta_f)$ .

### 3.2. Label Predictor

The label predictor  $G_c$  is composed of a fully connected layer. For the source domain data, it is equivalent to an ordinary supervised learning classifier. The parameters for the feature extractor  $G_f$  are refined by reducing the cross-entropy loss between the predicted and actual labels, which ensures the diagnostic precision of the source domain. At the same time, pseudo labels are output for unlabeled target domain samples, which are used for the subdomain alignment of local domain classifiers. If  $G_c$  can extract features that are invariant across both the source and target domains after transfer training, then the label predictor can be utilized for label prediction in the target domain. The extracted sample features are passed through the label classifier  $G_c$  with a parameter of  $\theta_c$  to obtain the predicted label  $\hat{y} = G_c(Z^s; \theta_c)$ . The loss for training on the source domain is as follows:

$$\begin{aligned}
 L_{cls} &= -\frac{1}{M} \left[ \sum_{i=1}^M J_y(\hat{y}_i, y_i^s) \right] \\
 &= -\frac{1}{M} \left[ \sum_{i=1}^M J_y(G_c(G_f(x_i^s; \theta_f); \theta_c), y_i^s) \right]
 \end{aligned} \tag{5}$$

### 3.3. Local Domain Discriminator

The local domain discriminator  $G_d^k$  is divided into multiple sub-discriminators. A single sub-discriminator is a binary classifier that comprises three fully connected layers. Each fully connected layer is accompanied by a batch normalization (BN) layer, and the rectified linear unit (ReLU) function is utilized to introduce nonlinearity. The number of sub-discriminators corresponds to the fault category and is used to align the conditional distribution between each category in the source and target domains. To be more specific, the local domain discriminator  $G_d^k$  can be partitioned into  $K$  subdomain discriminators. Each subdomain discriminator is responsible for aligning the features of source and target domain samples that are relevant to the  $K$ th class. The output of the label predictor  $G_c$  is the probability of each sample  $x_i$  category label belonging, and this probability is used to indicate the attention of the  $K$ th sub-domain discriminator to the sample. The loss function calculation formula of the local domain discriminator has been given in the first section.

### 3.4. Training Process

The MAJDAN objective function consists of three parts, which are the label predictor loss (Equation (5)); the MMD of the source and target domains (Equation (4)); and the local domain discriminator loss (Equation (2)). By combining (2), (4), and (5), the objective function can be expressed as follows:

$$L_{\text{total}}(\theta_f, \theta_c, \theta_d) = L_{\text{cls}} - \mu((1 - \lambda)L_d - \lambda L_{\text{MMD}}) \quad (6)$$

where  $\mu$  and  $\lambda$  are the tradeoff parameters of the objective function.

The training purpose of the model is to iteratively update the parameters  $\theta_f$ ,  $\theta_d$ , and  $\theta_c$ . In the MAJDAN, the diagnostic accuracy of the source domain samples is guaranteed by minimizing the supervised classification loss of the source domain. On this basis, the parameters  $\theta_f$  and  $\theta_c$  are optimized by using MMD to reduce the global distribution difference between the source domain and the target domain. Concurrently, the parameter  $\theta_d$  is adjusted by amplifying the domain classification loss of the domain discriminator via the Gradient Reversal Layer (GRL). This is performed to blur the distinction between the two domains, enabling the feature extractor to derive domain-invariant features. Algorithm 1 displays the MAJDAN training process.

---

#### Algorithm 1. MAJDAN model training.

---

Training procedure for the proposed MAJDAN algorithm

---

##### 1. Initialization

Input : data  $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ ,  $D_t = \{x_j^t\}_{j=1}^{n_t}$ ,

hyperparameter  $\lambda$  and  $\mu$ ,

learning rate  $\varepsilon$ ,

batch size  $b$ ,

initialization parameters  $\theta_f$ ,  $\theta_d$ ,  $\theta_c$ .

##### 2. Training

**for** each epoch **do**:

**for** each batch size **do**:

**for**  $i$  from 1 to  $b$  **do**:

Forward propagation

$$G_f(x_i) \leftarrow f(\theta_f, x_i)$$

$$G_c(G_f(x_i)) \leftarrow f(\theta_c, G_f(x_i))$$

$$G_d(G_f(x_i)) \leftarrow f(\theta_d, G_f(x_i))$$

Back propagation

using Equations (2), (4) and (5) calculate label classifier loss  $L_{\text{cls}}$ , MMD loss  $L_{\text{MMD}}$ , local domain discriminator loss  $L_d$ .

Update  $\theta_f$ ,  $\theta_d$ ,  $\theta_c$

$$\theta_f \leftarrow \theta_f - \varepsilon \left( \frac{\partial L_{\text{cls}}}{\partial \theta_f} - \mu \left( (1 - \lambda) \frac{\partial L_d}{\partial \theta_f} - \lambda \frac{\partial L_{\text{MMD}}}{\partial \theta_f} \right) \right)$$

$$\theta_c \leftarrow \theta_c - \varepsilon \left( \frac{\partial L_{\text{cls}}}{\partial \theta_c} + \mu \frac{\partial L_{\text{MMD}}}{\partial \theta_c} \right)$$

$$\theta_d \leftarrow \theta_d - \varepsilon \frac{\partial L_d}{\partial \theta_d}$$

**end**

##### 3. Testing

Using trained feature extractor and label classifier to predict the test set of the target domain.

---

## 4. Experimental Results

### 4.1. Dataset

The experimental dataset employed in this study originates from the University of Ottawa and comprises bearing vibration data collected under the condition of time-varying speed [42]. Figure 5 shows a photo and a diagram of the experimental platform. The data

are collected by the NI data acquisition board; the accelerometer measures the vibration data; and the encoder measures the rotational speed data. In this experiment, the bearing vibration data are selected as the dataset. The dataset collects bearing vibration signals of different health states under different time-varying speed conditions. The sampling frequency is 200 kHz, and the sampling time of each vibration signal is 10 s. Based on different health statuses, the dataset is categorized into three groups, as presented in Table 2.

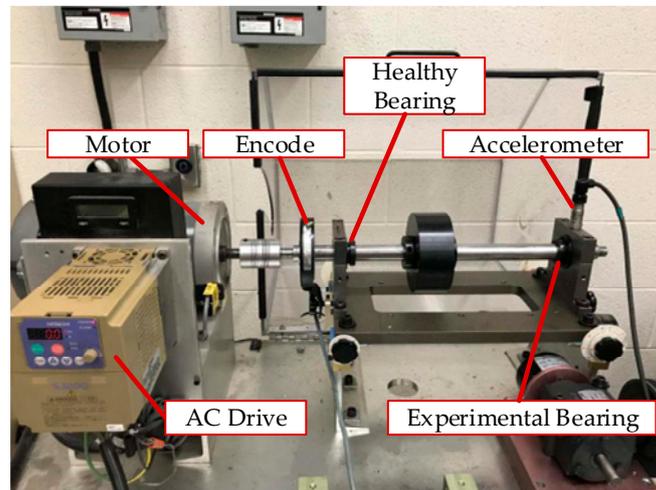


Figure 5. The Ottawa University bearing test bench.

Table 2. Fault status information from University of Ottawa bearing dataset.

Health Condition	Normal Condition	Inner Race Fault	Outer Race Fault
Label	H	I	O

This dataset also collects signals from four different variable speed conditions: increasing speed, decreasing speed, increasing then decreasing speed, and decreasing then increasing speed. Table 3 shows the labels of the four different working conditions in the experiment. In this experiment, transfer tasks were designed for the data of different working conditions, including 12 groups of single-working-condition transfer tasks: T0–T1, T0–T2, T0–T3, T1–T2, T1–T3, T2–T3, T1–T0, T2–T0, T3–T0, T2–T1, T3–T1 and T3–T2, and four groups of multiple-working-condition transfer tasks: T0–T1T2T3, T1–T0T2T3, T2–T0T1T3, and T3–T0T1T2.

Table 3. University of Ottawa bearing dataset working condition information.

Working Condition	Increasing Speed	Decreasing Speed	Increasing Then Decreasing Speed	Decreasing Then Increasing Speed
Label	T0	T1	T2	T3

Before the test, the data need to be processed in order to obtain a more ideal fault diagnosis result. This paper uses a 1DLCNN as a feature extractor, which can directly extract features from the original vibration signal. But the input data need to be normalized and cut first.

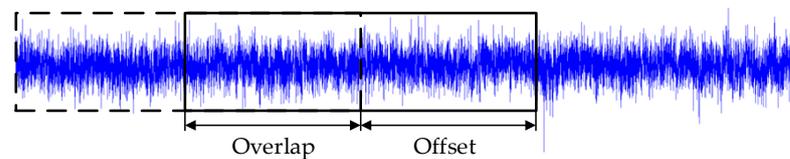
Normalization: To adapt the input data to the requirements of the neural network and mitigate the adverse effects of significant variations in magnitude among individual input

samples on the model's solution, before segmenting the vibration signal, the data must be normalized. The method used is norm normalization:

$$V' = \frac{V}{\|V\|_2} \quad (7)$$

where  $V$  denotes the original vibration signal, and  $V'$  represents the normalized signal.

**Cutting:** The original vibration signals from the bearings are continuously collected. To meet the training requirements of the neural network, the raw vibration signals are segmented. As shown in Figure 6, this study employs overlapping sampling segmentation to augment the dataset, thereby increasing the number of training samples and enhancing the generalization performance of the neural network [43]. The signal length of each acquisition is 4096, the overlap is 50%, and each working condition has 2900 samples after segmentation.



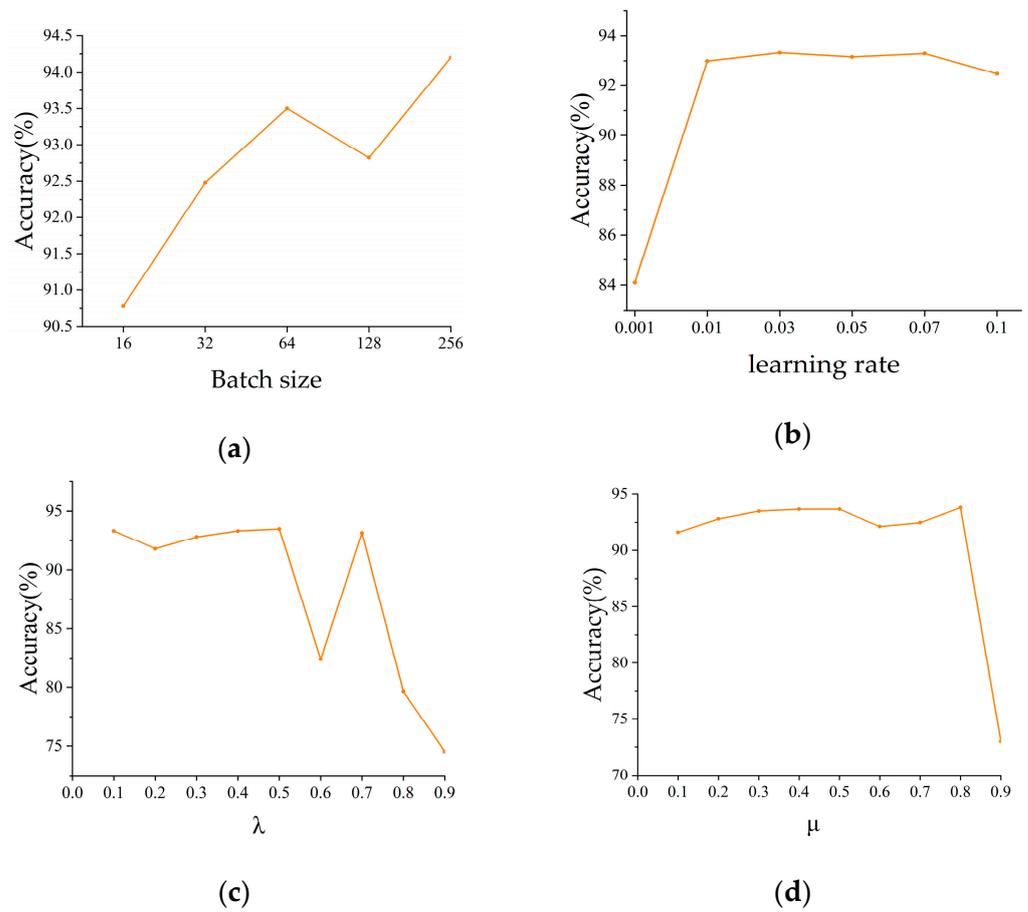
**Figure 6.** Overlap sampling method.

#### 4.2. Parameter Settings

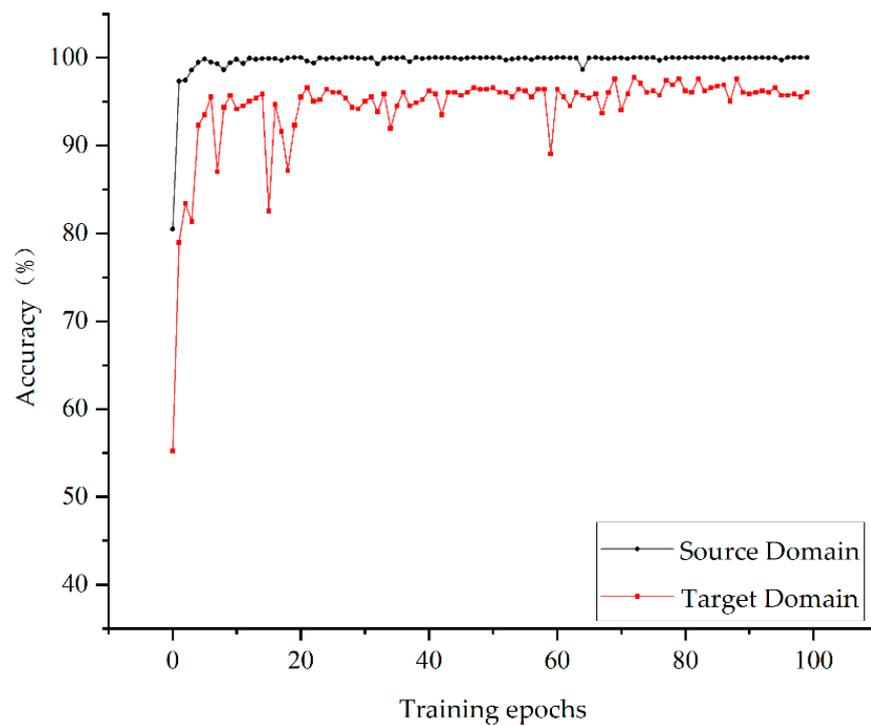
Hyperparameters can profoundly influence the model's performance. To enhance accuracy and the model's generalizability while curtailing training time, this study conducts experiments, taking the T0-T1 single operating condition transfer task as a case study, to examine the effects of hyperparameters on the model. The experiments were conducted by varying four different hyperparameters: batch size, initial learning rate,  $\lambda$ , and  $\mu$ . The model was trained for 50 epochs for each experiment. Figure 7 shows the comparison results with different hyperparameter settings. The results indicate that the values of  $\lambda$  and  $\mu$  have a significant impact on diagnostic accuracy, while batch size and initial learning rate have a relatively minor effect on accuracy. However, batch size and initial learning rate do affect the duration of the training process. After considering all factors, the final hyperparameter settings were chosen as follows: batch size of 64, initial learning rate of 0.03,  $\lambda$  set to 0.5, and  $\mu$  set to 0.5.

After determining the optimal range of hyperparameters, carry out transfer task experiments on T0, T1, T2, and T3 working conditions. All labeled samples from the source domain and 80% of the unlabeled samples from the target domain are selected for domain adaptation training to reduce the distribution difference between the two domains. The remaining 20% of the samples from the target domain are used to test the classification performance in the target domain after unsupervised domain adaptation.

Figure 8 illustrates the variation trend of accuracy in the source domain and target domain during the training of the MAJDAN model for the T0-T1 single-working-condition transfer task. The results indicate that the accuracy of the source domain samples reaches a stable state after five epochs of training, achieving a classification accuracy of approximately 100%. This is because the training of the source domain belongs to supervised learning, and the feature extractor used in this paper can extract data features well to classify labeled data. After 30 epochs of training in the target domain, the accuracy can reach about 96%, indicating that the MAJDAN model is capable of extracting domain-invariant features from unlabeled target domain data through domain adaptation techniques and accurately performing fault classification.



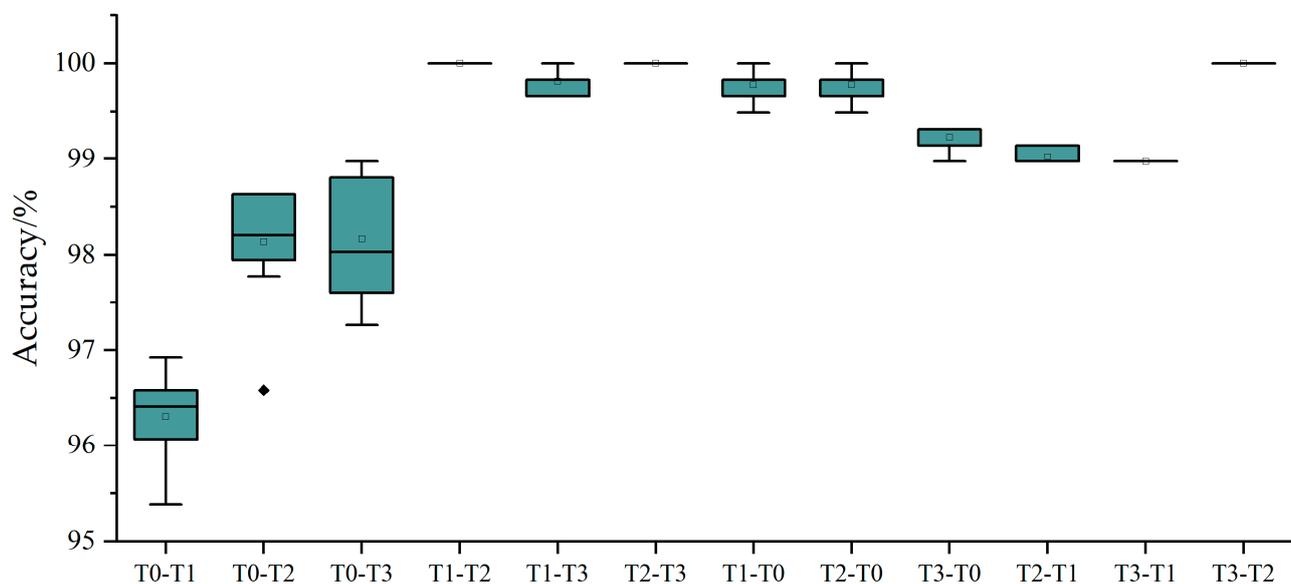
**Figure 7.** Accuracy with different hyperparameters. (a) Batch size. (b) Initial learning rate. (c) Hyperparameter  $\lambda$ . (d) Hyperparameter  $\mu$ .



**Figure 8.** Variation trend of diagnostic accuracy for different training times of T0–T1 transfer task.

#### 4.3. Single-Working-Condition Transfer Experiment

The experiment designed six groups of single-working-condition transfer tasks: T0–T1, T0–T2, T0–T3, T1–T2, T1–T3, and T2–T3. The source domain contains 2900 samples, while the target domain contributes 2300 samples for training and an additional 600 samples for testing. In each experiment, the model is trained for 50 epochs, and for each transfer task, the training is repeated 10 times. The accuracy of the 10 training runs is averaged to obtain the final result. The experimental results are shown in Figure 9, where the diagnostic accuracy of the target domain for the transfer tasks T1–T2, T2–T3, and T3–T2 reached 100%, and the diagnostic accuracy for the transfer tasks T1–T3, T1–T0, T2–T0, T3–T0, T2–T1, and T3–T1 exceeded 99%, and for T0–T2 and T0–T3, it was around 98%. The diagnostic accuracy of transferring from condition A to condition B is relatively lower compared to other migration tasks, but it is still above 96%. The experimental results confirm the effectiveness of the MAJDAN model.



**Figure 9.** Target domain diagnostic accuracy in different transfer tasks.

In practical situations, the collected signals mostly contain noise. In order to verify the anti-noise performance of MAJDAN, this paper introduces Gaussian white noise at varying signal-to-noise ratios (SNRs) to the bearing vibration signal. The definition of SNR is as follows:

$$SNR_{dB} = 10 \lg \left( \frac{P_{\text{signal}}}{P_{\text{noise}}} \right) \quad (8)$$

where  $P_{\text{signal}}$  is the power of the original signal, and  $P_{\text{noise}}$  is the power of the noise. Figure 10 shows the original vibration signal and the vibration signal with SNR = 50 dB, 45 dB, 40 dB, and 35 dB Gaussian noise added.

The same preprocessing and training methods as above are applied to the signal added with Gaussian white noise. Figure 11 shows the experimental results of the MAJDAN under various transfer tasks after adding noise with different SNRs to the signal.

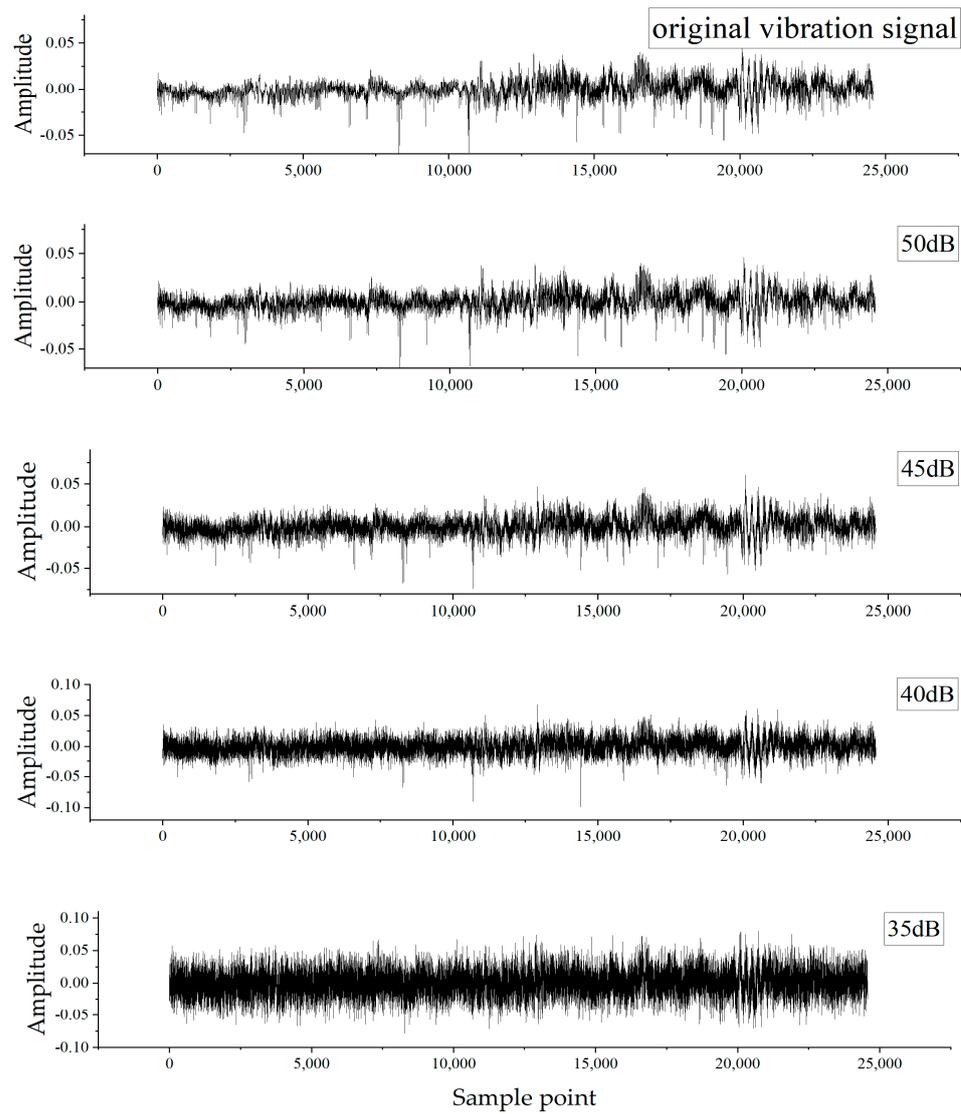


Figure 10. Vibration signals with different noise intensities.

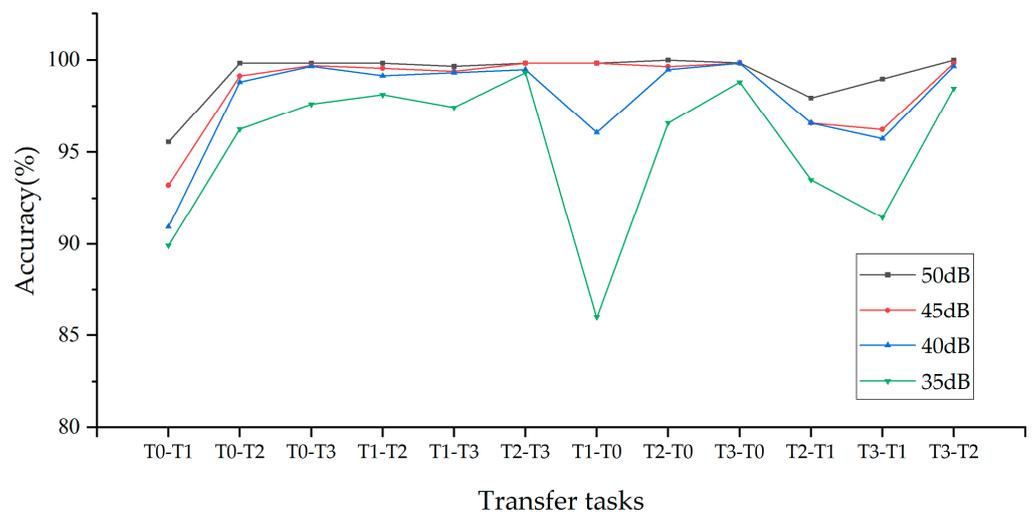


Figure 11. Accuracy of different noise intensities in each transfer task.

From the experimental results in Figure 11, it can be found that using the MAJDAN on the vibration signal with 35 dB noise can also make the diagnosis accuracy of each transfer task reach more than 85%, indicating that the network has a certain ability to resist noise interference. In order to further verify the superiority of the MAJDAN, this paper takes a sample with 45 dB noise as an example to compare it with other methods: 1D CNN, Domain Adversarial Neural Networks (DANNs) [44], CORAL [27], and deep subdomain adaptation network (DSAN) [45]. Each of the adaptive methods uses the same 1DLCNN structure as a feature extractor. The results of the comparison are shown in Table 4. The highest diagnostic accuracy for each transfer task is represented in bold. According to the results in the table, it can be found that, the network model used in this paper is more stable than other comparison models on various transfer tasks and has the highest accuracy on most transfer tasks. In a 45 dB noise environment, the average accuracy of the MAJDAN on all tasks is 6–15% higher than other comparison network models.

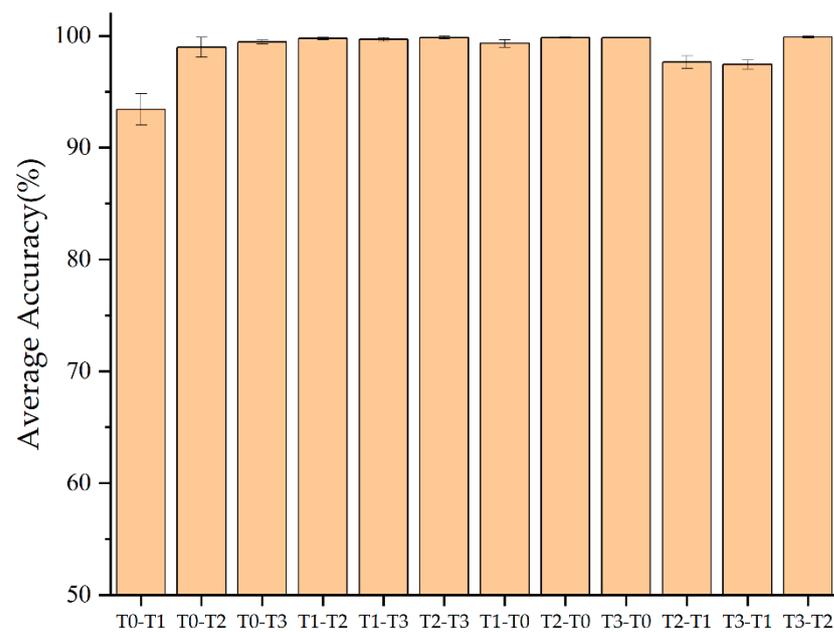
**Table 4.** Comparison results of single-working-condition transfer tasks.

	1D CNN	DANN	Coral	DSAN	MAJDAN
T0-T1	60.69%	72.44%	63.23%	53.71%	<b>93.21%</b>
T0-T2	63.21%	64.94%	65.22%	59.74%	<b>99.14%</b>
T0-T3	67.02%	74.15%	66.65%	99.58%	<b>99.69%</b>
T1-T2	99.32%	99.59%	99.55%	<b>99.62%</b>	99.55%
T1-T3	99.65%	<b>99.72%</b>	99.08%	99.58%	99.38%
T2-T3	99.48%	99.73%	99.81%	99.72%	<b>99.89%</b>
T1-T0	60.03%	88.89%	61.37%	99.65%	<b>99.83%</b>
T2-T0	65.64%	99.31%	64.95%	<b>99.82%</b>	99.65%
T3-T0	98.84%	99.82%	99.65%	99.67%	<b>99.83%</b>
T2-T1	96.14%	<b>98.12%</b>	96.75%	97.60%	96.58%
T3-T1	95.20%	<b>98.97%</b>	95.89%	98.81%	96.23%
T3-T2	99.74%	<b>99.83%</b>	99.15%	<b>99.83%</b>	<b>99.83%</b>
Average	83.75%	91.29%	84.28%	92.28%	<b>98.57%</b>

#### 4.4. Verification of Model Robustness

To ensure the robustness and reliability of the proposed algorithm, a 5-fold cross-validation method was employed for its evaluation. Specifically, the entire dataset was evenly segmented into five subsets. In each validation round, four of these subsets served as training data, with the remaining one used for testing. This approach ensured that the algorithm underwent training and testing on the entire dataset, providing a comprehensive evaluation.

After each validation round, the accuracy of the testing subset was recorded. Following the five validations, both the average and standard deviation of the diagnostic accuracy were computed to offer a holistic view of the algorithm's performance. This method allows for an assessment of not just the mean performance of the algorithm but also its stability across various data subsets. Figure 12 displays the average results and errors from the five validations. As can be seen from the figure, the standard deviations from the 5-fold cross-validation results are small, indicating that the model achieved similar performance across the five different data splits. This suggests that the model is unlikely to overfit to a specific data partition.



**Figure 12.** Average accuracy of 5-fold cross-validation.

#### 4.5. Multi-Working Condition Transfer Experiment

The effectiveness of the MAJDAN model in transferring single-working-condition fault diagnosis of bearings can be verified through the previous experiment. However, compared with the transfer between single working conditions, the transfer tasks of one-to-many working conditions are more practical and challenging. This means that in practical applications, it is often sufficient to collect labeled samples from a single operating condition and unlabeled samples from a mixture of various operating conditions to perform fault diagnosis on bearings or other mechanical equipment.

In order to verify the effect of the MAJDAN on one-to-many working condition transfer tasks, this section designs four groups of multi-working condition transfer tasks: T0-T1T2T3, T1-T0T2T3, T2-T0T1T3, and T3-T0T1T2. The training method and result-obtaining method are the same as in the single case. Table 5 shows the diagnostic accuracy of different methods and their average for each transfer task. According to the training results, the performance of the MAJDAN in the one-to-many transfer task is also more stable than other comparison methods. The diagnostic accuracy on the four transfer tasks can reach more than 97%, and the average accuracy is the highest compared with other methods.

**Table 5.** Comparison results of one-to-many working condition transfer tasks.

	1D CNN	DANN	Coral	DSAN	MAJDAN
T0-T1T2T3	62.15%	64.23%	66.44%	96.40%	<b>98.12%</b>
T1-T0T2T3	88.35%	80.66%	87.84%	88.52%	<b>99.82%</b>
T2-T0T1T3	87.03%	98.63%	99.12%	98.97%	<b>99.14%</b>
T3-T0T1T2	87.15%	97.83%	97.60%	<b>98.29%</b>	97.78%
Average	81.17%	85.34%	87.75%	95.55%	<b>98.71%</b>

#### 4.6. Performance of MAJDAN on Other Dataset

In addition to the bearing dataset from the University of Ottawa, we also selected the bearing dataset from Case Western Reserve University (CWRU) [46] to verify the performance of MAJDAN. The CWRU dataset has four working conditions: 1797 rpm/0 hp (R0), 1772 rpm/1 hp (R1), 1750 rpm/2 hp (R2), and 1730 rpm/3 hp (R3). The accelerometer is used to sample the fault data of the drive end bearing, and the sampling frequency is

12 KHz. Table 6 shows the 10 health states contained in the bearing dataset, including the normal state and three fault states, and each fault state is divided into three damage degrees.

**Table 6.** Fault status information from CWRU dataset.

Health Condition	Fault Size	Label
Outer race fault	7 mils	O-7
	14 mils	O-14
	21 mils	O-21
Inner race fault	7 mils	I-7
	14 mils	I-14
	21 mils	I-21
Ball fault	7 mils	B-7
	14 mils	B-14
	21 mils	B-21
Normal		N

The processing method of the dataset is the same as before, and each working condition participating in the training has 2000 samples. A total of 80% of the dataset as the target domain is used for training, and the rest is used for testing. Table 7 shows the diagnostic accuracy of each transfer task using MAJDAN and other models. Compared with 1DCNN without an adaptive strategy, the algorithm proposed in this paper has obvious advantages in diagnosis accuracy. Compared with other adaptive methods, DANN and CORAL also showed superiority. The average accuracy of MAJDAN's cross-working condition diagnosis on the CWRU dataset is close to 100%, which verifies the effectiveness of the model on different datasets.

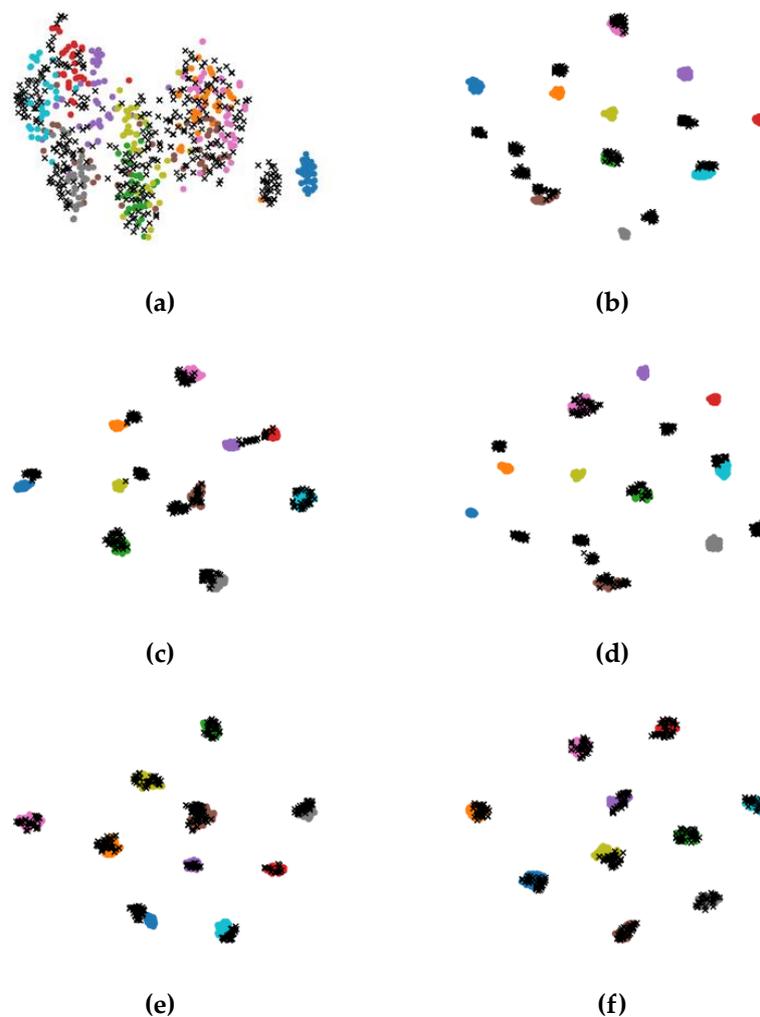
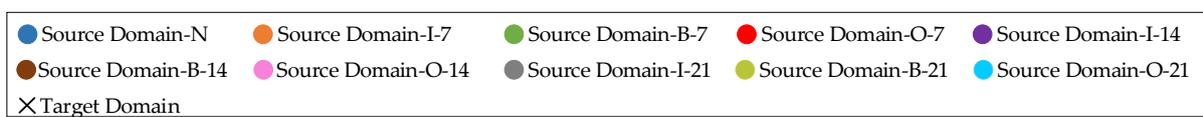
**Table 7.** Comparison results of CWRU dataset.

	1D CNN	DANN	CORAL	DSAN	MAJDAN
R0-R1	94.63%	99.75%	96.75%	<b>100%</b>	99.87%
R0-R2	85.44%	<b>100%</b>	88.66%	<b>100%</b>	<b>100%</b>
R0-R3	77.50%	78.25%	78.25%	<b>99.87%</b>	<b>99.87%</b>
R1-R2	97.25%	99.75%	97.50%	99.25%	<b>100%</b>
R1-R3	71.75%	99.75%	71.83%	<b>99.75%</b>	<b>99.75%</b>
R2-R3	94.75%	99.75%	94.25%	<b>99.75%</b>	<b>99.75%</b>
R1-R0	85.50%	97.42%	98.25%	99.00%	<b>99.66%</b>
R2-R0	92.75%	87.38%	87.87%	<b>99.31%</b>	98.25%
R3-R0	83.20%	78.55%	63.25%	<b>97.25%</b>	95.75%
R2-R1	93.25%	93.76%	87.00%	95.44%	<b>97.75%</b>
R3-R1	76.75%	77.25%	77.50%	96.25%	<b>96.87%</b>
R3-R2	92.00%	96.66%	96.75%	98.00%	<b>99.87%</b>
Average	87.06%	92.36%	86.49%	98.66%	<b>98.95%</b>

#### 4.7. Feature Visualization

In order to understand the impact of different models more intuitively on sample features, the R0-R3 transfer task is selected for feature visualization. In this study, the t-distributed Stochastic Neighbor Embedding (t-SNE) technique was employed to reduce the dimensionality of sample features to 2D for visualization. t-SNE constructs two probability distributions expressing the similarity between sample points in high-dimensional and

low-dimensional space, respectively. It uses gradient descent to learn and optimize these distributions, aiming to make them as similar as possible. This ensures that the distance relationships between data points are preserved after dimensionality reduction. Figure 13 shows the graphical representation of untrained raw data features and features after training with different models following dimensionality reduction. It is observed that the source domain samples in (b)–(f) all achieve excellent classification results, which also proves the effectiveness of the feature extractor and label predictor. In Figure 13b, the 1DCNN model undergoes no transfer training, leading to significant dissimilarity between the source domain and target domain sample distributions. Consequently, this mismatch results in poor diagnostic performance. The source and target domain sample features in (a) and (d) are also not well matched, while the features of the two domains in (e) and (f) are very close. But in some categories, feature clusters of (f) appear to be tighter and better matched than (e).



**Figure 13.** t-SNE feature visualization. (a) Features without training. (b) 1DCNN. (c) DANN. (d) CORAL. (e) DSAN. (f) MAJDAN.

## 5. Conclusions

This paper designs a MAJDAN model for cross-condition bearing fault diagnosis, integrating MMD and multi-adversarial networks, to address the challenges of varying operating conditions and difficulties in acquiring labeled data in practical scenarios. The model was validated on bearing vibration datasets for transfer learning tasks, and the results confirmed the effectiveness of the proposed MAJDAN model in this paper. Through the experiments in this paper, the following conclusions are drawn:

(1) The MAJDAN model uses a 1DLCNN for feature extraction, which can effectively extract the features of the original data. Make the model reduce training costs and data processing steps while ensuring classification accuracy.

(2) The MAJDAN model combines MMD and multi-adversarial networks for extracting domain-invariant features, which can simultaneously align the marginal and conditional distributions of the source and target domains. It shows excellent performance in single-condition transfer and one-to-many case transfer tasks.

(3) The MAJDAN model also exhibited strong performance in transfer tasks involving bearing vibration data with added Gaussian noise. Compared to other transfer learning methods, the proposed model demonstrates more stable diagnostic accuracy across multiple transfer tasks and achieves higher average accuracy. It exhibits excellent robustness against noise.

**Author Contributions:** Conceptualization, Z.C. and H.C.; methodology, Z.C. and Z.A.; validation, Z.C., H.C. and Z.A.; formal analysis, H.C.; investigation, J.W.; resources, H.C.; data curation, Z.C. and J.W.; writing—original draft preparation, Z.C.; writing—review and editing, Z.C., H.C. and Z.A.; visualization, Z.C.; supervision, H.C.; project administration, H.C.; funding acquisition, H.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the project Development of Liquid Cargo and Electromechanical Simulation Operation System for LNG Ship, grant number CBG3N21-3-3; National Key R&D Program of China, grant number 2022YFB4301400.

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

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. This data can be found here: <https://data.mendeley.com/datasets/v43hmbwxpm/1> (accessed on 11 March 2023).

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

## References

1. Kankar, P.K.; Sharma, S.C.; Harsha, S.P. Fault diagnosis of ball bearings using machine learning methods. *Expert Syst. Appl.* **2011**, *38*, 1876–1886. [[CrossRef](#)]
2. Guo, L.; Li, N.; Jia, F.; Lei, Y.; Lin, J. A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing* **2017**, *240*, 98–109. [[CrossRef](#)]
3. Lu, Q.; Yang, R.; Zhong, M.; Wang, Y. An improved fault diagnosis method of rotating machinery using sensitive features and RLS-BP neural network. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 1585–1593. [[CrossRef](#)]
4. Chen, S.; Yang, R.; Zhong, M.; Xi, X.; Liu, C. A random forest and model-based hybrid method of fault diagnosis for satellite attitude control systems. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 3279453. [[CrossRef](#)]
5. Wei, M.H.; Yang, J.W.; Yao, D.C.; Wang, J.H.; Hu, Z.S. Fault diagnosis of bearings in multiple working conditions based on adaptive time-varying parameters short-time Fourier synchronous squeeze transform. *Meas. Sci. Technol.* **2022**, *33*, 124002. [[CrossRef](#)]
6. Chen, C.Z.; Sun, C.C.; Yu, Z.; Nan, W. Fault diagnosis for large-scale wind turbine rolling bearing using stress wave and wavelet analysis. In Proceedings of the 8th International Conference on Electrical Machines and Systems (ICEMS 2005), Nanjing, China, 27–29 September 2005; pp. 2239–2244.
7. Hu, A.J.; Xiang, L. Selection principle of mathematical morphological operators in vibration signal processing. *J. Vib. Control* **2016**, *22*, 3157–3168. [[CrossRef](#)]
8. Fernandez-Canti, R.M.; Blesa, J.; Tornil-Sin, S.; Puig, V. Fault detection and isolation for a wind turbine benchmark using a mixed Bayesian/Set-membership approach. *Annu. Rev. Control* **2015**, *40*, 59–69. [[CrossRef](#)]

9. 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](#)]
10. Zhao, R.; Yan, R.Q.; Chen, Z.H.; Mao, K.Z.; Wang, P.; Gao, R.X. Deep learning and its applications to machine health monitoring. *Mech. Syst. Signal Process.* **2019**, *115*, 213–237. [[CrossRef](#)]
11. Zhao, B.; Zhang, X.; Li, H.; Yang, Z. Intelligent fault diagnosis of rolling bearings based on normalized CNN considering data imbalance and variable working conditions. *Knowl. Based Syst.* **2020**, *199*, 105971. [[CrossRef](#)]
12. Jiao, J.; Zhao, M.; Lin, J.; Liang, K. A comprehensive review on convolutional neural network in machine fault diagnosis. *Neurocomputing* **2020**, *417*, 36–63. [[CrossRef](#)]
13. Chen, H.; Meng, W.; Li, Y.; Xiong, Q. An anti-noise fault diagnosis approach for rolling bearings based on multiscale CNN-LSTM and a deep residual learning model. *Meas. Sci. Technol.* **2023**, *34*, 045013. [[CrossRef](#)]
14. Ding, Y.; Jia, M.; Miao, Q.; Cao, Y. A novel time-frequency Transformer based on self-attention mechanism and its application in fault diagnosis of rolling bearings. *Mech. Syst. Signal Process.* **2022**, *168*, 108616. [[CrossRef](#)]
15. Zhou, K.; Diehl, E.; Tang, J. Deep convolutional generative adversarial network with semi-supervised learning enabled physics elucidation for extended gear fault diagnosis under data limitations. *Mech. Syst. Signal Process.* **2023**, *185*, 109772. [[CrossRef](#)]
16. Zhao, M.; Jiao, J.; Lin, J. A Data-Driven Monitoring Scheme for Rotating Machinery Via Self-Comparison Approach. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2435–2445. [[CrossRef](#)]
17. Zhang, R.; Tao, H.; Wu, L.; Guan, Y. Transfer Learning With Neural Networks for Bearing Fault Diagnosis in Changing Working Conditions. *IEEE Access* **2017**, *5*, 14347–14357. [[CrossRef](#)]
18. Li, C.; Zhang, S.; Qin, Y.; Estupinan, E. A systematic review of deep transfer learning for machinery fault diagnosis. *Neurocomputing* **2020**, *407*, 121–135. [[CrossRef](#)]
19. Xiao, D.; Huang, Y.; Zhao, L.; Qin, C.; Shi, H.; Liu, C. Domain Adaptive Motor Fault Diagnosis Using Deep Transfer Learning. *IEEE Access* **2019**, *7*, 80937–80949. [[CrossRef](#)]
20. Zhou, T.; Han, T.; Droguett, E.L. Towards trustworthy machine fault diagnosis: A probabilistic Bayesian deep learning framework. *Reliab. Eng. Syst. Saf.* **2022**, *224*, 108525. [[CrossRef](#)]
21. Liu, Y.; Wang, Y.; Chow, T.W.S.; Li, B. Deep Adversarial Subdomain Adaptation Network for Intelligent Fault Diagnosis. *IEEE Trans. Ind. Inform.* **2022**, *18*, 6038–6046. [[CrossRef](#)]
22. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [[CrossRef](#)]
23. Xiao, N.; Zhang, L.; IEEE Comp, S.O.C. Dynamic Weighted Learning for Unsupervised Domain Adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021; pp. 15237–15246.
24. Yu, F.; Wang, D.; Chen, Y.; Karianakis, N.; Shen, T.; Yu, P.; Lymberopoulos, D.; Lu, S.; Shi, W.; Chen, X.; et al. SC-UDA: Style and Content Gaps aware Unsupervised Domain Adaptation for Object Detection. In Proceedings of the 22nd IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 4–8 January 2022; pp. 1061–1070.
25. Wang, C.; Chen, D.; Chen, J.; Lai, X.; He, T. Deep regression adaptation networks with model-based transfer learning for dynamic load identification in the frequency domain. *Eng. Appl. Artif. Intell.* **2021**, *102*, 104244. [[CrossRef](#)]
26. Borgwardt, K.M.; Gretton, A.; Rasch, M.J.; Kriegel, H.-P.; Schoelkopf, B.; Smola, A.J. Integrating structured biological data by Kernel Maximum Mean Discrepancy. *Bioinformatics* **2006**, *22*, E49–E57. [[CrossRef](#)] [[PubMed](#)]
27. Sun, B.C.; Saenko, K. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In Proceedings of the 14th European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 443–450.
28. Zellinger, W.; Grubinger, T.; Lughofer, E.; Natschläger, T.; Saminger-Platz, S. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv* **2017**, arXiv:1702.08811.
29. Ganin, Y.; Lempitsky, V. Unsupervised domain adaptation by backpropagation. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1180–1189.
30. Li, X.; Zhang, W.; Ding, Q.; Sun, J.-Q. Multi-Layer domain adaptation method for rolling bearing fault diagnosis. *Signal Process.* **2019**, *157*, 180–197. [[CrossRef](#)]
31. Wang, Y.; Sun, X.J.; Li, J.; Yang, Y. Intelligent Fault Diagnosis with Deep Adversarial Domain Adaptation. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 3035385. [[CrossRef](#)]
32. Qin, Y.; Yao, Q.W.; Wang, Y.; Mao, Y.F. Parameter sharing adversarial domain adaptation networks for fault transfer diagnosis of planetary gearboxes. *Mech. Syst. Signal Process.* **2021**, *160*, 107936. [[CrossRef](#)]
33. Khan, S.A.; Kim, J.M. Rotational speed invariant fault diagnosis in bearings using vibration signal imaging and local binary patterns. *J. Acoust. Soc. Am.* **2016**, *139*, EL100–EL104. [[CrossRef](#)]
34. Si, J.; Shi, H.M.; Chen, J.C.; Zheng, C.C. Unsupervised deep transfer learning with moment matching: A new intelligent fault diagnosis approach for bearings. *Measurement* **2021**, *172*, 108827. [[CrossRef](#)]
35. Sun, G.D.; Yang, X.; Xiong, C.Y.; Hu, Y.; Liu, M.Y. Rolling Bearing Fault Diagnosis Based on Time-Frequency Compression Fusion and Residual Time-Frequency Mixed Attention Network. *Appl. Sci.* **2022**, *12*, 4831. [[CrossRef](#)]
36. Yuan, H.D.; Chen, J.; Dong, G.M. Machinery fault diagnosis based on time-frequency images and label consistent K-SVD. *Proc. Inst. Mech. Eng. Part C-J. Mech. Eng. Sci.* **2018**, *232*, 1317–1330. [[CrossRef](#)]
37. Sandler, M.; Howard, A.; Zhu, M.L.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 4510–4520.

38. 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. [[CrossRef](#)] [[PubMed](#)]
39. 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.
40. Pei, Z.Y.; Cao, Z.J.; Long, M.S.; Wang, J.M. Multi-Adversarial Domain Adaptation. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence/30th Innovative Applications of Artificial Intelligence Conference/8th AAAI Symposium on Educational Advances in Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; AAAI: Washington, DC, USA, 2018; pp. 3934–3941.
41. Gretton, A.; Borgwardt, K.M.; Rasch, M.J.; Schölkopf, B.; Smola, A. A kernel two-sample test. *J. Mach. Learn. Res.* **2012**, *13*, 723–773.
42. Huang, H.; Baddour, N. Bearing vibration data collected under time-varying rotational speed conditions. *Data Brief* **2018**, *21*, 1745–1749. [[CrossRef](#)]
43. Jin, G. *Research on End-to-End Bearing Fault Diagnosis Based on Deep Learning under Complex Conditions*; University of Science and Technology of China: Hefei, China, 2020; p. 3.
44. 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.
45. 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.* **2020**, *32*, 1713–1722. [[CrossRef](#)]
46. Case Western Reserve University Bearing Dataset. Available online: <https://engineering.case.edu/bearingdatacenter/apparatus-and-procedures> (accessed on 21 May 2023).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.