A Deep Transfer Learning-Based Network for Diagnosing Minor Faults in the Production of Wireless Chargers

Wang, Y., Li, W. & Zhu, H.

DOI 10.3390/app132011514
ESSN 2076-3417

Publisher: MDPI

applied sciences

Article

A Deep Transfer Learning-Based Network for Diagnosing Minor Faults in the Production of Wireless Chargers

Yuping Wang [1,*], Weidong Li [1,2,3,*] and Honghui Zhu [1]

[1] School of Transport and Logistics Engineering, Wuhan University of Technology,
Wuhan 430070, China; 115526@whut.edu.cn
[2] Faculty of Engineering, Environment and Computing, Coventry University, Coventry CV1 5FB, UK
[3] School of Mechanical Engineering, University of Shanghai for Science and Technology,
Shanghai 200444, China
* Correspondence: yuping.wang@whut.edu.cn (Y.W.); weidongli@usst.edu.cn (W.L.)

Abstract: Wireless charger production is critical to energy storage, and effective fault diagnosis of bearings and gears is essential to ensure wireless charging performance with high efficiency, high tolerance to misalignment, and thermal safety. As minor faults are usually difficult to detect, timely diagnosis and detection of minor faults can prevent the fault from worsening and ensure the safety of wireless charging systems. Diagnosing minor faults in bearings and gears with data is a useful but difficult task. To achieve a satisfactory diagnosis of minor faults in the production of wireless charging systems related to the mechanical system that produces wireless charging devices, such as robot arms, this paper proposes a deep learning network based on CNN and LSTM (DTLCL). The method uses deep learning network, model-based transfer learning and range adaptation technology. First, a deep neural network is built to extract significant fault features. Second, the deep transfer network is initialised using model-based transfer learning with a good starting point. Finally, range adaptation using the maximum mean discrepancy between the features learned from the source and target ranges is realised by a multi-layer adaptive technology. The effectiveness of the method was verified using actual measurement data. The training time is 19 s, and the accuracy exceeds 94.5%. The explanation results show that the proposed DTLCL method provides higher accuracy and robust identification of smaller errors compared to the current combination of integrated and single non-transmission models. Due to its data-driven nature, the DTLCL method could be used for fault diagnosis of bearings and gears, which would further promote the application process of wireless charging.

Keywords: wireless charging applications; transfer learning; minor fault diagnosis; maximum mean discrepancy; wireless charging equipment

check for updates

Citation: Wang, Y.; Li, W.; Zhu, H. A Deep Transfer Learning-Based Network for Diagnosing Minor Faults in the Production of Wireless Chargers. *Appl. Sci.* **2023**, *13*, 11514. https://doi.org/10.3390/app132011514

Academic Editor: Alessandro Lo Schiavo

Received: 21 August 2023
Revised: 16 October 2023
Accepted: 18 October 2023
Published: 20 October 2023

1. Introduction

Bearings and gears are one of the most important components in the manufacture of wireless chargers. They affect the transmission efficiency, alignment error tolerance, charging power matching and thermal reliability of wireless charger performance [1–3]. For example, the shaft of the wireless charger is a key component, and the installation of the shaft is related to the service life and experience of the wireless charger. To enable more comfortable use and prevent the charger from being difficult to pull out, the modern charger is equipped with a rotating design, as the shaft causes scratches that reduce the rotation efficiency. In this regard, bearings and gears are incorporated into the shaft to improve the flexibility and efficiency of rotation and reduce friction. To improve the stability and durability of the wireless charger, it is also necessary to add a suitable heat dissipation function so that the heating mechanism of the charger's fan can include a heat dissipation plate and the connecting bearing and gear [4]. In this context, fault diagnosis of bearings and gears in wireless charging applications is necessary and useful.

Deep Learning (DL) is a promising tool for automatic feature learning due to its deep architecture. It is widely used in natural language processing, state monitoring, speech recognition and other fields. In recent years, according to the rapid development of artificial intelligence technology, deep learning algorithms such as Stacked Auto-Encoder (SAE), Deep Belief Network (DBN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and Long Short Memory (LSTM) have been widely used in fault diagnosis. For example, Jia et al. [5] stacked multiple AEs to extract features from raw bearing vibration signals without the need for professional technicians. However, due to the complexity of the original signal data, the AE method, which uses mean square error (MSE) as a loss function, is not very robust, so the performance is not good. Shao et al. [6] proposed an improved depth model AE which is a combination of DAE and comparative AE (CAE). Chen et al. [7] proposed feature extraction methods for bearing fault diagnosis: DBM and DBN. Li et al. [8] conducted experiments on gear and bearing fault data and verified that the selection of low-level feature domains has a profound effect on the deep statistical features of DBM. Shao et al. [9] proposed PSO to optimise the DBN structure for fault diagnosis. Janssens et al. [10] proposed feature learning based on CNN using two sensors to detect vibration signals. Guo et al. [11] proposed a hierarchical CNN method with an adaptive learning rate for bearing fault classification. Ding et al. [12] transformed the fault diagnosis problem into an image recognition problem. Wang et al. [13] used 1D data with a CNN model. The parameters of the model were determined using the PSO algorithm. Many results show that DL has strong scalability and generalisation capability compared to previously used machine learning algorithms (ML) such as Logistic Regression (LR), k-Nearest Neighbour (k-NN) and Support Vector Machine (SVM) and does not require manual feature extraction [14–16]. However, DL-enabled methods still have some limitations, such as: (1) source and target domains are evenly distributed; (2) the target domain has enough error data. Moreover, it is still difficult to meet both of the above requirements simultaneously. In practice, the accuracy of the features of the Deep Learning-based fault diagnosis model is inevitably affected by the number of fault samples and low quality. Moreover, the effectiveness of fault diagnosis cannot be guaranteed. Therefore, it is necessary to develop effective models to solve the problem of micro-fault diagnosis of label-free data in wireless charging devices.

Transfer Learning (TL) is an efficient machine learning method that can use the knowledge in the corresponding source domain to solve the above challenges [17,18]. This is because TL is a method for transferring data or features from the source domain to the target domain and improves model performance in the target domain with fewer data or features by using the source domain with more data or features. In general, TL methods are divided into model-based transfer learning (MTL), feature-based transfer learning (FTL), instance-based transfer learning (ITL) and relation-based transfer learning (RTL). FTL and MTL are two of the most popular methods. In MTL, the initialisation of the target domain model is usually pre-trained with data from the source domain. MTL is currently used in fault diagnosis and gives good results [19,20]. FTL can change the properties of source and target domains by a domain adaptive method to identify a common potential space. If there is little or no labelling data for the target domain, FTL can be used [21]. The domain-adaptive method often consists of the maximum mean discrepancy (MMD). MMD is the distribution distance between the computation of the source domain and the target domain [22]. FTL includes shallow methods such as Transfer Component Analysis and Joint Distribution Adaptation, and deep methods such as Domain Confusion, Deep Adaptation Network, Domain Adversarial Neural Network and Deep CORAL (D- CORAL). Currently, FTL is used in the diagnosis of bearing faults. For example, based on the FTL method, Sapkota et al. [23] assumed that there are some overlaps between the source domain and the target domain and proposed a Structural Corresponding Learning (SCL) method. Nevertheless, the robustness of each model is sometimes low. Sanodiya et al. [24] proposed training different transformation matrices for the source and target domains to achieve the goal of transfer learning. Based on the MTL method, Li et al. [25] proposed the

TransEMDT method, which uses a decision tree to build a robust behaviour recognition model based on the labelled data. The RTL method is poorly researched and discussed in only a few articles [26]. There are other transfer learning methods. Yaroslav Ganin et al. [27] proposed the DANN method to add a confrontation mechanism to neural network training. Bousmalis et al. [28] from Google Brain extended DANN by proposing a DSN network. TL in smart manufacturing for fault diagnosis is still in its infancy [29]. The existing methods cannot be transferred between DL models created at different defect levels and mixed defects, so early micro-defect diagnosis and multiple defect diagnosis are not solved and are challenging. Due to limited space, this article focuses on early micro-defect diagnosis.

For this reason, CNN has a good feature extraction capability. In the task of temporal sequence, LSTM can solve the problem of vanishing gradient caused by the gradual reduction in gradient without consideration. Considering the advantages of both methods, this study proposes a Deep Transfer Learning Network (DTLCL) based on CNN and LSTM for bearing micro-fault diagnosis with unlabelled or sparsely labelled data in wireless charging applications. The proposed method is based on DL, MTL, FDL and domain adaptation. First, a deep neural network (DNN) based on CNN and LSTM is built and pre-trained to learn transferable features, labelling the source domain data as significant error data. Second, by initialising the model of the target domain, MTL obtains a relatively good starting point. The network structure and number of neurons in each layer of the target domain model are identical to that of the DNN. Finally, FTL is used to learn invariant features in the source and target domains through Deep Domain Adaptation (DDA) [30]. The calculation of MMD loss by a Gaussian and linear kernel function can measure the distribution distance more effectively. The kernel MMD selection is designed to use the validation accuracy of the target model to assign an appropriate weighted voting (WV), which is a popular combination strategy. In this way, DTLCL trained with labelled data of significant faults can be used to effectively predict unlabelled or poorly labelled micro-fault diagnostic data. Case studies of varying complexity have shown that the DTLCL method has advantages over any base model and other existing TL methods. The case studies also illustrate the relevance of the DTLCL method for wireless charging in a real environment with signal interference and noise. The summary of the contributions is as follows: a combined learning method with inheritance depth transfer based on CNN, LSTM and weighted tuning algorithms is proposed for the adaptive diagnosis of minor faults in rolling bearings of wireless chargers.

This study makes the following contributions: (1) This method makes use of Deep Learning and Transfer Learning. Moreover, DNN autonomously extracts features from unprocessed vibration data in wireless charger manufacturing, which provides excellent flexibility without the need to manually convert and extract features; (2) MTL is used to prepare the source domain data to initialise the target domain model and give it a solid foundation; (3) A linear combination of Gaussian kernels of WV is used to create the MMD to better assess the differences between the source domain and the target domain; (4) Compare the case studies with Deep Learning without transfer and the existing traditional transfer learning. The effectiveness of the method was verified using actual measurement data. The training time is 19 s, and the accuracy exceeds 94.5%. The explanation results show that the proposed DTLCL method can be more accurate and solid than the current combination of integrated combinations and single models without transfer or transfer in identifying small errors.

The rest of this paper is structured as follows. Section 2 introduces the basic theory of DL and TL. Section 3 explains the framework for deriving the DTLCL method. Section 4 explains the experiment and analyses the experimental results. Section 5 concludes and discusses limitations, possible applications and difficulties.

2. Basic Theory

In this section, certain TL and DL-related notations are introduced to explicitly express the problem to be solved.

2.1. CNN and LSTM

CNN has excelled in many areas because it can learn from older agents through multiple layers. A CNN consists of input, output and several hidden layers. In the hidden layer, there are convolutional layers, pooling layers and fully linked layers [31]. First, convolution is performed by inputting a one-dimensional or two-dimensional input and a convolution kernel. Second, after convolution, a non-linear activation function is added. Third, to reduce the size of the output feature map, pooling is usually performed. Fourth, after a series of iterations of convolution and subsampling, fully connected layers are used for classification. Finally, a softmax function is performed. Furthermore, backpropagation is used to optimise the parameters of the CNN by minimising the classification loss.

In a sense, the recurrent neural network (RNN) is the most detailed model [32]. RNNs can only address problems with short-term dependence. A unique RNN that can handle both short-term and long-term dependence problems is the LSTM, as time-series data, signals from smart manufacturing make the LSTM a promising tool for micro-fault diagnosis.

2.2. MMD-Based TL

To solve the problem of micro-error diagnosis for unlabelled data or data with few labels, transfer learning is introduced as follows. Normally, the data from the source and target domains do not come from the same distributions. Kernel MMD is a non-parametric measure of distribution discrepancy. To measure distribution discrepancy more effectively, most studies have used kernel MMD [33]. The formulation of MMD can be defined as follows:

$$D(S,T) = \frac{1}{Ns^2}\sum\nolimits_{i,j=1}^{Ns} k(S_i,S_j) - \frac{2}{NsNt}\sum\nolimits_{i,j=1}^{Ns,Nt} k(S_i,T_j) + \frac{1}{Nt^2}\sum\nolimits_{i,j=1}^{Nt} k(T_i,T_j) \tag{1}$$

where Ds is source domain, and Dt is target domain. $S = \left\{y^i \middle| S_i, i = 1,2,\ldots N_s\right\}$ is source domain dataset, N_s is the total sample count, y is its actual label, $S_i = \left(s^i_{1,} s^i_{2,} s^i_{3,} \ldots s^i_{p,}\right)$ is the ith sample, and p is the dimensionality of the sample. Similarly, $T = \left\{T_j, j = 1,2,\ldots N_t\right\}$ is the target domain dataset without the label, in which N_t is the total number of the sample, $T_j = \left(t^j_{1,} t^j_{2,} t^j_{3,} \ldots t^j_{q,}\right)$ is the jth sample. k is used to depict a kernel, such as the Gauss kernel.

However, the selection of the MMD kernel by a single Gaussian kernel in [34,35] is challenging as it affects the feature mapping performance. Moreover, in certain research, accuracy and resilience are not good [36,37]. Therefore, it is of great importance to develop new methods to solve this problem.

3. The Proposed Methodology

In this study, a DTLCL method is proposed for the diagnosis of micro-faults with unlabelled data. Figure 1 shows the flowchart of the proposed system, which mainly consists of three parts, namely DNN-based CNN and LSTM, MTL and DTLCL design. DNNs are used to discover features from numerous notable fault samples. A supervised backpropagation algorithm is used to fine-tune and optimise DNN parameters. Limited label samples are used to optimise and fine-tune the DNN parameters by minimising the loss function. They obtain the model DNNs and the model parameters trained with many samples with severe errors. The network structure of the model DNNt is the same as that of the DNNs, and the number of neurons in each layer is also the same. MTL can initialise the DNN used as the target model with a good starting point. This is because it is often used to pre-train the target model using data from the source domain. It has recently been used in fault diagnosis and has achieved excellent results [38,39]. DTLCL was developed by combining DNN, MTL, FTL and domain adaptation to realise transfer learning from the model of significant faults (DNNs) to the model of smaller faults (DNNt). The three-layer adaptation of the kernel MMD enables domain adaptation. The choice of a kernel is difficult. Therefore, a new comprehensive metric has been developed to assist WV in assigning appropriate voting weights for kernel MMD selection. Finally, DTLCL policies can be adaptively created to examine both source and target domain characteristics. In particular, DTLCL with two different kernels can improve the diversity of DNNs of the target model and learn features with a small discrepancy between domains, which is challenging for a single kernel.

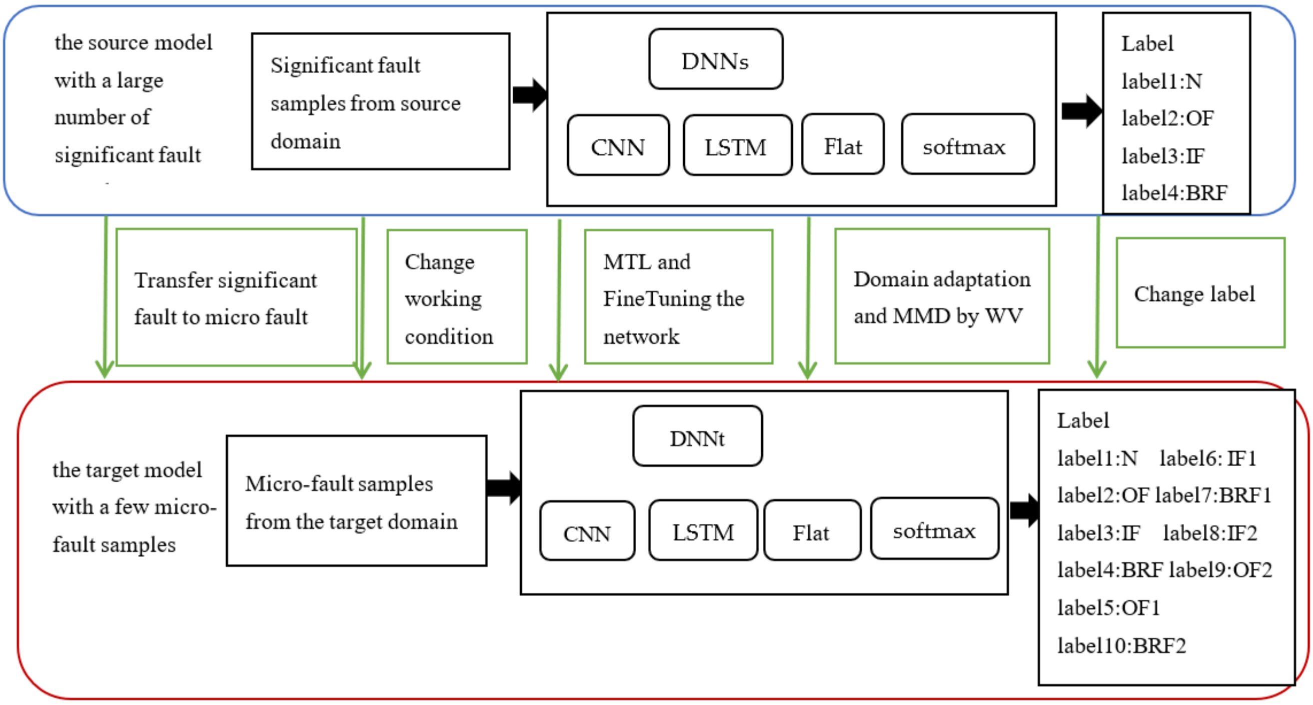

the source model with a large number of significant fault
Significant fault samples from source domain
DNNs
CNN
LSTM
Flat
softmax
Label
label1:N
label2:OF
label3:IF
label4:BRF
Transfer significant fault to micro fault
Change working condition
MTL and FineTuning the network
Domain adaptation and MMD by WV
Change label
the target model with a few micro-fault samples
Micro-fault samples from the target domain
DNNt
CNN
LSTM
Flat
softmax
Label
label1:N label6: IF1
label2:OF label7:BRF1
label3:IF label8:IF2
label4:BRF label9:OF2
label5:OF1
label10:BRF2

Figure 1. Flowchart of the DTLCL approach.

3.1. DNN Construction-Based Deep Learning

In this section, a CNN combination of LSTM models is used as a Deep Neural Network to construct a Deep Transfer Network (DTLCL) because of its excellent feature learning capability.

3.1.1. Raw Data Pre-Processing

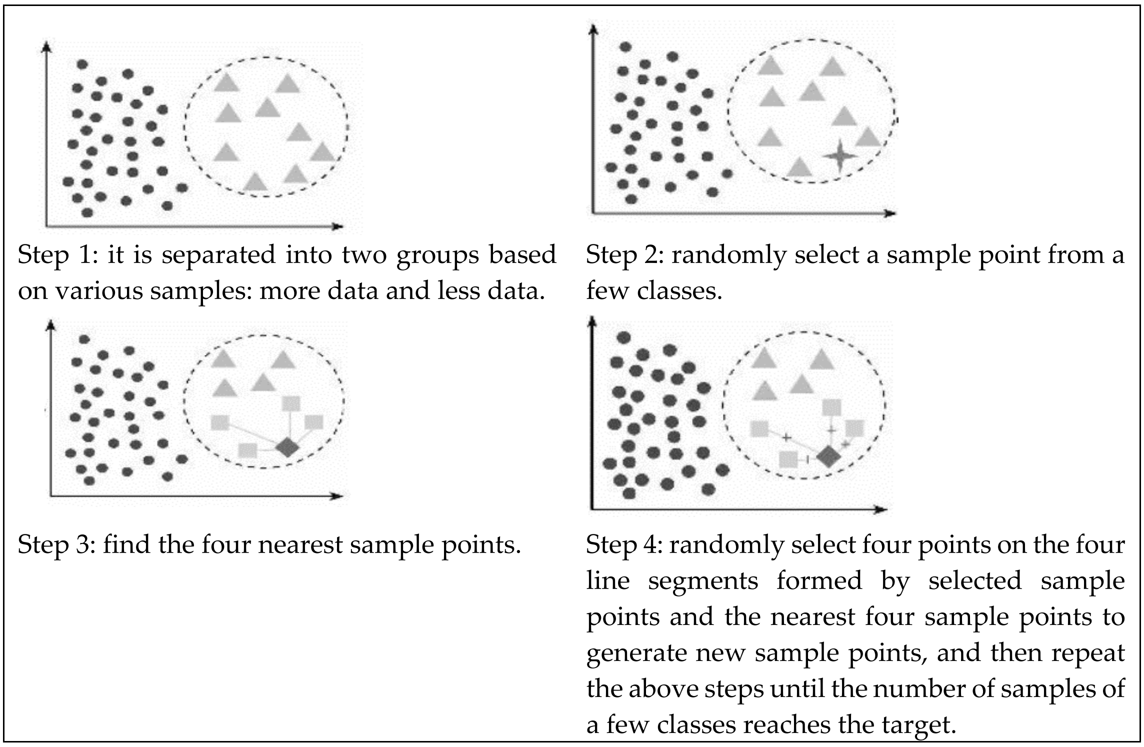
First, significant and minor errors are pre-processed from the acquired vibration signals. In this study, a new method of overlapping sampling by sliding windows is proposed. The sampling point is 2048, the step length of deviation (S) is 28, the standard deviation normalises the data, and then the data is coded in one pass. With this method, the dataset gets N 620,544 data, the number of training samples is N-(L-S) and the dataset is divided into training set, verification set and test set in the ratio of 7:2:1 after processing.

Second, training dataset X_s with many significant fault samples and training dataset X_i with only a few minor fault samples are obtained.

To eliminate the negative effects caused by large differences in the dimensions of the characteristic variables. Standardisation of the data is particularly important. Min–max normalisation is used, which can be described as follows:

$$x^* = \frac{x - min}{max - min} \tag{2}$$

where x, x^* is pre-conversion and converted value. max, min is the maximum and minimum value of the original data. If the sample size is uneven, the model may perform well in the training dataset but not in the test dataset. For this purpose, Synthetic Minority Oversampling Technique (SMOTE) is used, which is a synthesis of some classes of oversampling techniques to make better use of the data [40,41]. Figure 2 shows the principle of SMOTE, by assuming that some classes are oversampled four times.



Step 1: it is separated into two groups based on various samples: more data and less data.

Step 2: randomly select a sample point from a few classes.

Step 3: find the four nearest sample points.

Step 4: randomly select four points on the four line segments formed by selected sample points and the nearest four sample points to generate new sample points, and then repeat the above steps until the number of samples of a few classes reaches the target.

Figure 2. The principle of SMOTE.

3.1.2. Design DNN Construction

Recent research shows that it is impossible to transfer deep features and classifiers from the source domain to the target domain using a pure deep high-level model [42,43]. Therefore, we develop the integrated method of DNN to learn the classifiers and transferable invariant features of source and target domains by combining the LSTM and CNN models, as shown in Figure 3. The proposed model includes three convolutional modules, one LSTM module, one shallow layer, three dense layers and one softmax classifier. One-dimensional convolution was chosen because the vibration signal contains only one-dimensional data. In certain processes, initially adding a batch back to the convolution layer and pooling layer may cause the input to be pulled back to the convolution layer to obey the standard normal distribution. This can prevent the gradient from disappearing while further speeding up convergence and training speed. Then, adding the LSTM network after the pooling layer, can solve the problem of long-term dependence or gradient explosion to better refine the properties. Finally, to prevent overfitting and improve the generalisation ability, the dropout layer is added to the full connection layer. The structure of the CNN consists of Convolution, BatchStandardisation and Maximum Pooling. The input of Convolution is (None, 2048, 1), the output is (None, 128, 16), and the input of BatchStandardisation is (None, 128, 16), the output is (None, 128, 16), the input of MaxPooling1D is (None, 128, 16), the output is (None, 64, 16). After the CNN structure has been traversed three times, the LSTM network structure is added, whose input is (None, 16, 32) and output is (None, 16, 4). Table 1 shows the selection of architectural parameters of the DNN model by grid search and k-fold cross-validation [44].

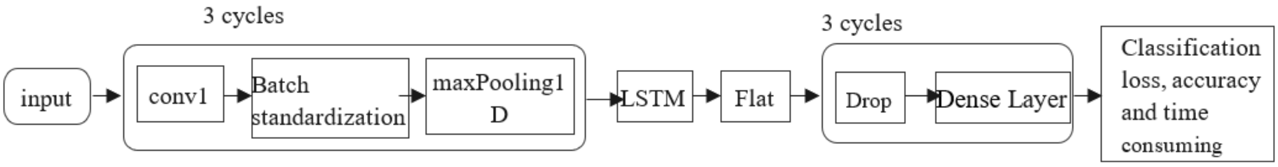


Figure 3. The architecture of DNN.

Table 1. Architecture parameters' selection of the DNN model.

Layers	Parameters	Activation	Output Size
Input	/	/	(None, 2048, 1)
conv1D	Filters:16, kernel_size: 64, strides: 16	relu	(None, 128, 16)
BatchNormalization	/	/	(None, 128, 16)
MaxPooling1D	pool_size: 2	relu	(None, 64, 16)
conv1D	Filter: 16, kernel_size: 64, strides: 16	relu	(None, 64, 32)
BatchNormalization	/	/	(None, 64, 32)
MaxPooling1D	pool_size: 2	relu	(None, 32, 32)
conv1D	Filters: 16, kernel_size: 64,strides: 16	relu	(None, 32, 32)
BatchNormalization	/	/	(None, 32, 32)
MaxPooling1D	pool_size: 2	relu	(None, 16, 32)
LSTM	recurrent_activation: hard_sigmoid	tanh	(None, 16, 32)
Flatten	/	/	(None, 512)
Dropout	0.3		(None, 512)
Dense1	/	relu	(None, 256)
Dense2	/	relu	(None, 128)
Dense3	/	relu	(None, 32)
Classifier	kernel_regularizer: l1(1 × 10^{-4})	softmax	(None, 4)

A deep neural network fault diagnosis model for significant faults is created and trained, which can be described as follows:

$$DNN_s \ = Feedforward(f_{s1}, f_{s2}, \ldots, f_{sN}) \tag{3}$$

$$[DNN_s \ , \theta_{s'}] = train(DNN_s, X_s, \theta_s) \tag{4}$$

DNN_s consists of CNN and LSTM. f_{sj} represents the number of neurons in the jth buried layer of DNN_s, $j = 1, 2, \ldots., N$.

First, as shown in Equation (4), the model DNN_s from significant fault is trained. X_s is a training dataset from significant faults. $\theta_s = \{\theta_{s1}, \theta_{s2}, \ldots, \theta_{sN}\}$ is an initial set of parameters for the network $DNN_s.\theta_{si} = \{W_{si},\ b_{si}\}$ represents the set of parameters of bias and weight matrix of the input layer and hidden layer in DNN_s, which are initialized randomly.

Second, Layer-by-layer training updates DNN_s parameters $\theta_{s'}$.

Third, abstract features $F_{sN} = \sigma(W_{sN} \ldots (\sigma(W_{s2}(\sigma(W_{s1}X_s + b_{s1}) + b_{s2})) + \ldots + b_{sN})$ are captured.

Fourth, using F_{sN} as input data, the softmax classifier is trained to update and obtain the softmax parameters $\theta_{ss'}$.

Fifth, DNN_s parameters are fine-tuned and optimised using a supervised back-propagation algorithm. DNN_s is optimised by minimizing loss function with labelled dataset, which can be described as follows:

$$loss = -\sum_{i=1}^{outputsize} y_i.logp_i \tag{5}$$

where p_i is actual output probability, and y_i is expected output.

As the mode includes CNN and LSTM, the loss function is enhanced as follows:

$$loss = -\sum_{i=1}^{outputsize} y_i.logp_i + \lambda \tag{6}$$

with

$$\lambda = \frac{F\max}{F_{cnn} + F_{lstm}}, F\max = \max(F_{cnn}, F_{lstm}), \lambda \subset (0.01, 1) \tag{7}$$

$$R = \frac{TP}{TP + FN} \tag{8}$$

$$P = \frac{TP}{TP + FP} \tag{9}$$

$$F = \frac{2 \times P \times R}{R + P} \tag{10}$$

where *TP*, *FN*, *FP* stand for true positive, false negative, and false positive, F_{cnn} and F_{lstm} represent the comprehensive index of CNN and LSTM model in training, respectively.

Finally, DNN_s and parameters $T = \{\theta_s{}', \theta_{ss}{}'\}$ are trained and obtained by a large number of significant fault samples.

For minor faults, a deep neural network fault diagnosis model is established and trained, which can be described as follows:

$$DNN_t = Feedforward(f_{t1}, f_{t2}, \ldots, f_{tN}) \tag{11}$$

$$f_{ti} = f_{si}, i = 1 \ldots N, \tag{12}$$

3.2. Transfer from DNN_s to DNN_t

3.2.1. Transfer of Network Parameters

Because the input dimensions of DNN_s and DNN_t are the same, the quantity of neurons in the buried layer is similar. The network parameters from the first layer through the nth layer of DNN_s trained by many significant fault samples are $\{\theta_{s1}{}', \theta_{s2}{}', \ldots, \theta_{sN}{}'\}$. The network settings for the corresponding layer of DNN_t from micro-fault samples are $\{\theta_{t1}{}', \theta_{t2}{}', \ldots, \theta_{tN}{}'\}$, which can be described as

$$\theta_{ti}{}' = \theta_{si}{}', i = 1...N \tag{13}$$

3.2.2. Domain Adaptation

The main focus of deep network adaptation is the specific number of specific layers and measurement standards for adaptive adaptation. Figure 4 shows the process of the adaptive approach. In this method, the initial model mainly consists of the following layers. Convolutional, standard and maximally pooled functions are run three times, followed by the addition of LSTM networks, flat layers with dropout processing, several fully connected layers and the addition of a current number of target sets. The model parameters are initialised randomly, and the target model is trained with the target dataset. The feature transfer capability decreases dramatically in the higher layers as the domain discrepancy increases when deep features eventually transition from universal to particular through the network. The network adaptation technique used in this work is MMD. The dense1, dense2 and dense3 layers modify the distribution of the learned features. The bottleneck layer of the transfer model is the layer where the features are extracted. The first three layers of the classifier are complemented by a layer that uses an adaptive measurement criterion. One measurement criterion is the loss function. The loss function consists of the multi-class cross entropy loss and the MMD. Between the source and target domains, the MMD with multiple kernels is used. The loss of the DTLCL model after optimisation is as follows:

$$loss = -\sum_{i=1}^{n} y_i.logp_i + \lambda D \tag{14}$$

$$D = \alpha D\left(f_1^s, f_1^t\right) + \beta D\left(f_2^s, f_2^t\right) + \mu D\left(f_3^s, f_3^t\right) \tag{15}$$

$$\alpha + \beta + \mu = 1 \tag{16}$$

where $\lambda = \frac{\mathrm{Fmax}}{\mathrm{F_{cnn}} + \mathrm{F_{lstm}}}$, α, β, μ are coefficients, $\mathrm{f}_1^s, \mathrm{f}_1^t, \mathrm{f}_2^s, \mathrm{f}_2^t, \mathrm{f}_3^s, \mathrm{f}_3^t$ are the output of the layer dense1, dense2 and dense3 for both source and target domain, D is a multilayer MMD which is the linear combination of Gaussian kernels.

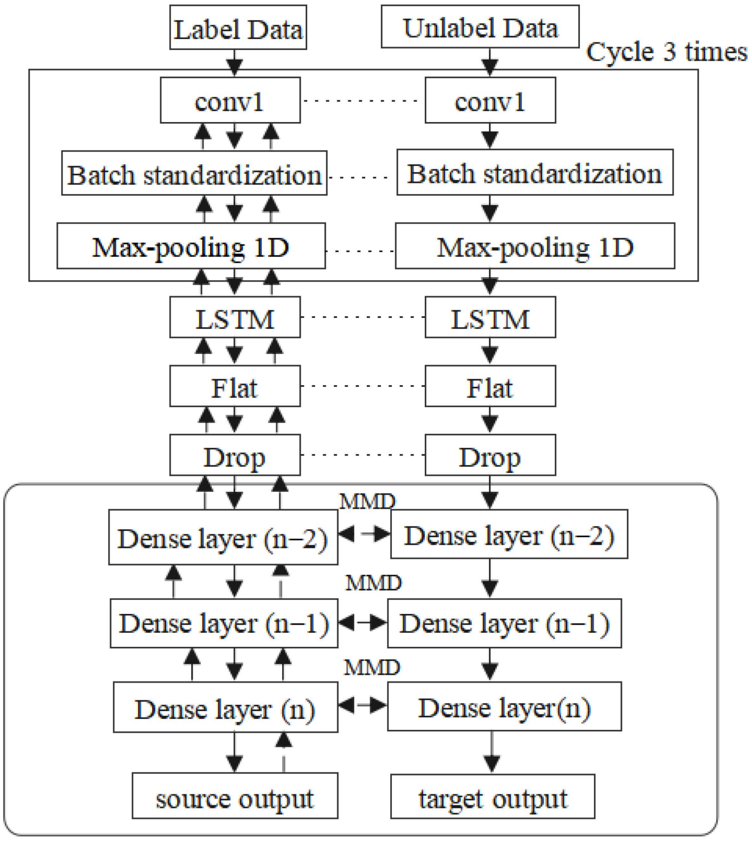


Figure 4. The adaptive method.

3.2.3. Weighted Voting for Kernel MMD Selection

The conclusion is that kernel MMD selection for transfer learning is crucial to improve the accuracy and generalisation ability of transfer learning. Weighted voting is a widely used combination strategy in which the weight of each model is determined based on its performance [44]. Suppose there are two base kernels, including Gaussian kernels. The weights of the two base kernels are w = {W1, W2}. WV takes into account the performance differences between the base kernels and gives a higher weight to the kernel with a higher accuracy. The weights of the kernels calculated with WV can be expressed as follows:

$$w_i = \frac{Model\_Accuracy_i}{\sum_{i=1}^{2} Model\_Accuracy_i} \tag{17}$$

where

$$\sum_{i=1}^{2} Model\_Accuracy_i = 1, w_i \geq 0 \tag{18}$$

and $Model\_Accuracy_i$ reflects the overall accuracy of the validation of ith kernel.

3.3. Transfer of Softmax Layer Parameters

Since minor faults and significant faults are under different working conditions, they may have different fault types, so the dimensions of softmax classifier are different. The transfer strategy of the softmax layer is to transfer only fault types common between minor and significant faults under different working conditions. Other different fault types are initialized randomly. $\theta_{ss'}$ is the parameter of softmax from DNN_s, $\theta_{ts'}$ is parameter of softmax from DNN_t, which can be described as follows:

$$\theta_{ss'} = \begin{bmatrix} w_{11}\, w_{12}\, w_{13} \ldots w_{1H} \\ w_{21}\, w_{22}\, w_{23} \ldots w_{2H} \\ w_{31}\, w_{32}\, w_{33} \ldots w_{3H} \\ \cdots \\ w_{n1}\, w_{n2}\, w_{n3} \ldots w_{nH} \end{bmatrix} \tag{19}$$

$$\theta_{ts'} = \begin{bmatrix} w_{11}\, w_{12}\, w_{13} \ldots w_{1H}\ \beta_{1(H+1)}\beta_{1(H+2)} \cdots\ \beta_{1F} \\ w_{21}\, w_{22}\, w_{23} \ldots w_{2H}\ \beta_{2(H+1)}\beta_{2(H+2)} \cdots\ \beta_{2F} \\ w_{31}\, w_{32}\, w_{33} \ldots w_{3H}\ \beta_{3(H+1)}\beta_{3(H+2)} \cdots\ \beta_{3F} \\ \cdots \\ w_{n1}\, w_{n2}\, w_{n3} \ldots w_{nH}\ \beta_{n(H+1)}\beta_{n(H+2)} \cdots\ \beta_{nF} \end{bmatrix} \tag{20}$$

where it is assumed that the status of significant faults is divided into H classes, and the status of minor faults is divided into F classes, $H < F$. H classes of significant faults is the previous class F of minor faults. β stands for random initialization.

3.4. Model Evaluation

Once the model is built, the advantages and disadvantages of the model must be evaluated using a comprehensive evaluation index (F), Recall (R) and Precision (P). The index of P and R is simple. F combines the indicators P and R presented in Section 3.1.2.

4. Experiment Verification

4.1. Experimental Platform Construction and Data Description

As an indispensable part of smart manufacturing for wireless charging, the stability of the wireless charging system is directly affected by the condition of the bearing. In this section, the validity and significance of the DTLCL method are tested using two different bearing datasets (dataset A and dataset B). Table 2 provides further details on the datasets.

Table 2. Data details of the bearing data set.

Dataset	Load/hp	Speed of Rotation/RPM	Fault Conditions	Samples Size
A	0/1/2/3	1797/1772 1750/1730	N/RF//IF/OF	1200 × 4
B	0/1/2	500/1000/1425	N/RF/IF/OF GPF/GBTF	200 × 6

Case Western Reserve University (CWRU) provided dataset A [45]. The data files are in MATLAB format. In Figure 5, the experimental platform is depicted. There are four fault states in dataset A, namely normal fault (N), outer circle fault (OF), inner circle fault (IF) and rolling element fault (RF). Each fault type has three different fault diameters (0.007 inch, 0.014 inch and 0.021 inch). The vibration signals are recorded at a sampling rate of 12 kHz. The test bearings are loaded with four different motor speeds (1797 rpm, 1772 rpm, 1750 rpm and 1730 rpm) and motor loads (0 HP, 1 HP, 2 HP and 3 HP), which are considered four different working conditions. Each data set includes 1200 samples, 300 samples per condition and 100 samples per fault.

Dataset B comes from the Intelligent Manufacturing Research Institute of Wuhan University of Technology and is shown in Figure 6a. The platform is driven by a SEW DRE100M4/BE5/HF/V/FI motor. The motor has the following specifications: 2.2 kW output power, 1425 RPM rated speed and 4 Nm rated torque. The roller bearing is a 6209-deep groove ball bearing with dimensions of 45 mm inside, 85 mm outside and 19 mm in width. Four vibration sensors record the vibration signals of the bearing at different positions of the motor drive side, the fan side and the pedestal, respectively, at different loads and speeds and use them as experimental data to diagnose bearing faults with a sampling frequency of 12 kHz. Figure 6b shows the positions of the fault point sensors.

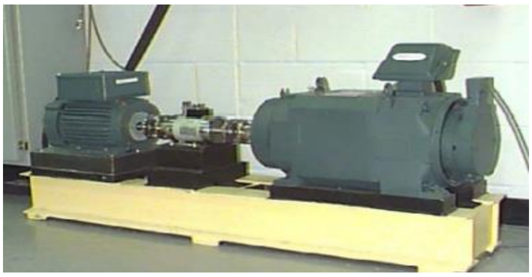


Figure 5. The experimental platform from CWRU.

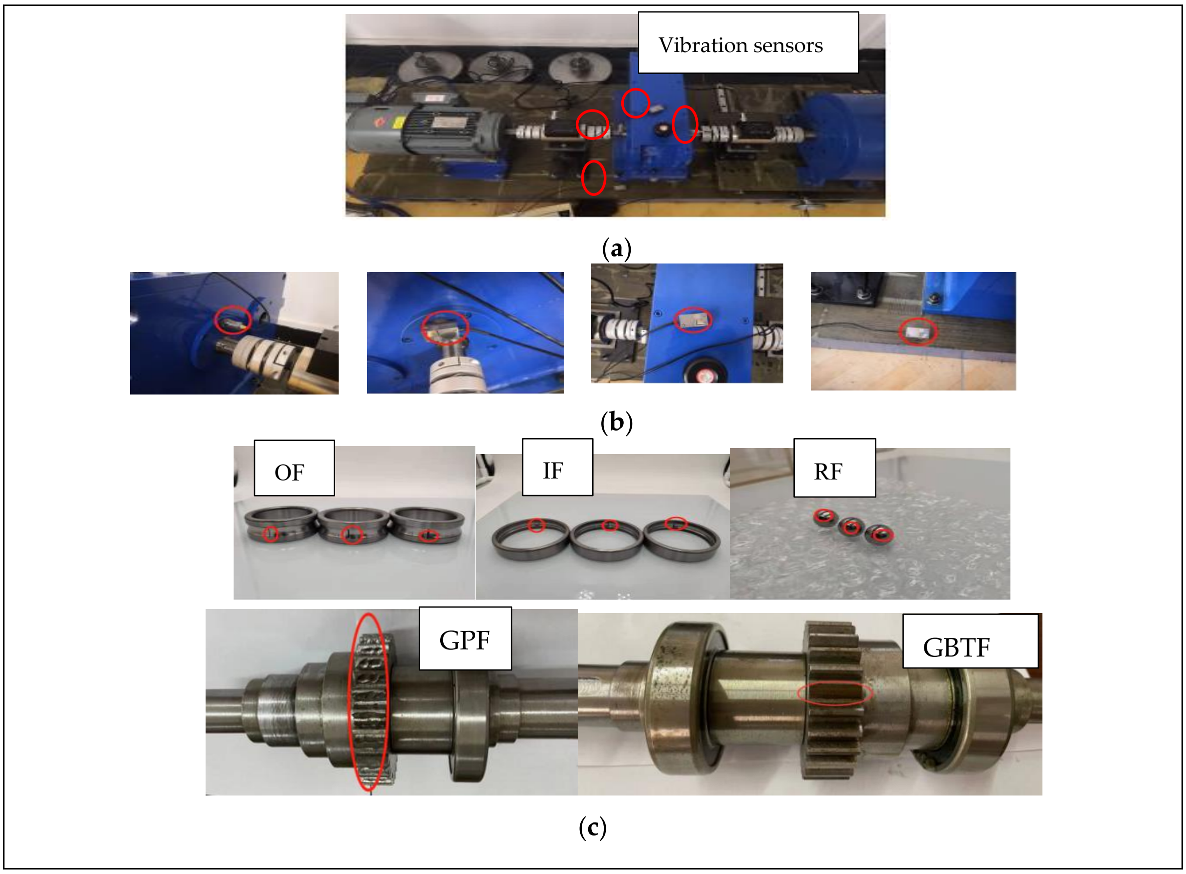
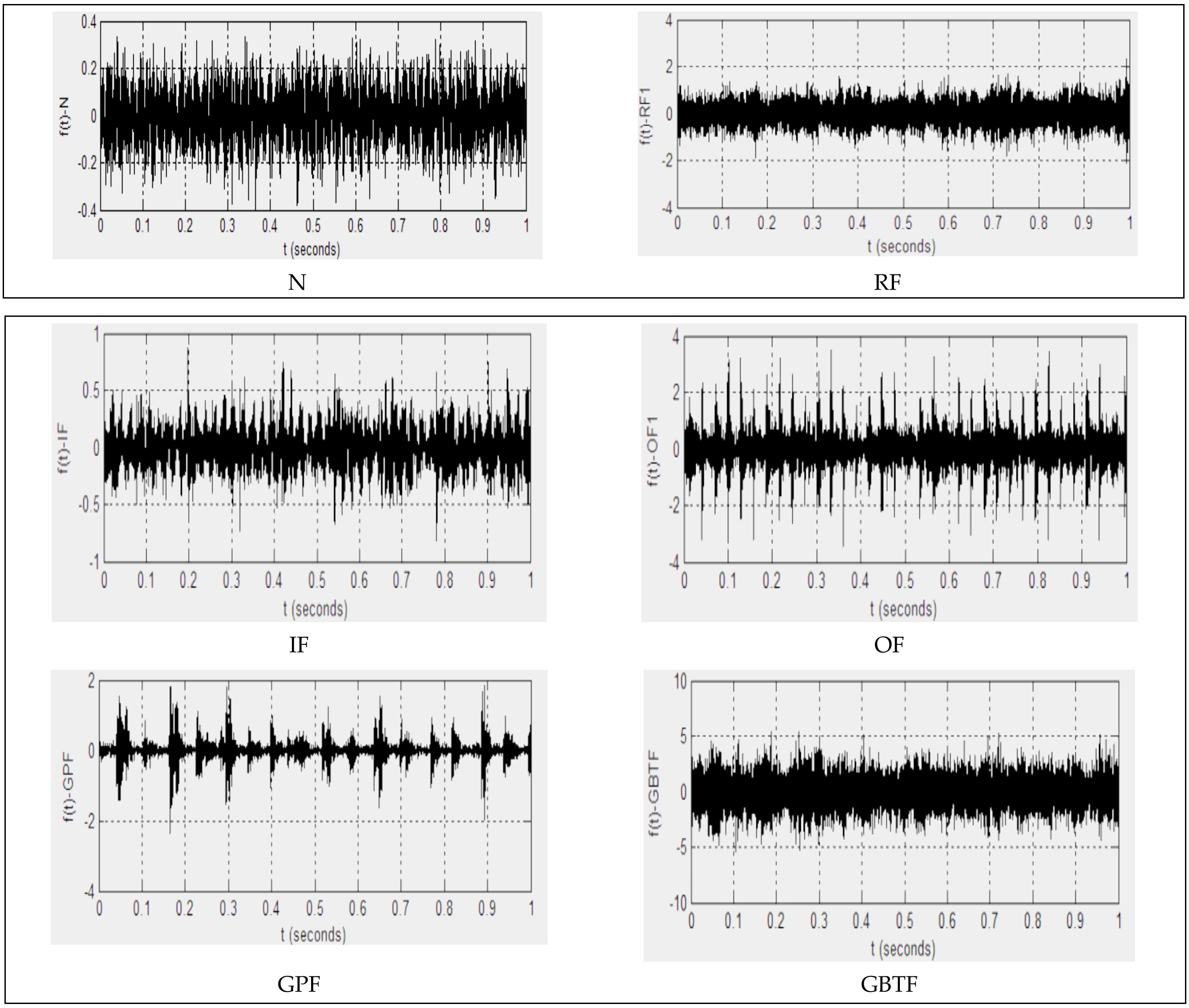


Figure 6. Experimental platform deployment. (**a**) Experimental platform. (**b**) 4 sensors in different positions. (**c**) Faulty parts of rolling bearings.

Dataset B is divided into six operating states with no failure and varying degrees of failure severity. Light failure is 0.3 mm, medium failure is 0.6 mm and severe failure is 0.9 mm. The rated speed is 500 rpm, 1000 rpm and 1425 rpm, and the corresponding input current is 0.0 A, 0.1 A and 0.2 A. Each sample consists of continuously recorded 048 points. A total of 1200 samples were collected under six different operating conditions. There are 840, 240 and 120 samples for the training, verification and test data sets, respectively. The six fault conditions are: normal fault (N), outer circle fault (OF), inner circle fault (IF), pitting fault (GPF), rolling element fault (RF) and broken tooth fault (GBTF). For a mild fault, the fault diameter was set at 0.0018 inches, for a moderate fault at 0.0036 inches, and for a severe fault at 0.0054 inch. Table 2 shows the engine speed and load of different faults in data set A and data set B. Figure 7 shows typical original data collected from four vibration sensors at different positions in Figure 6a. The acquired data is then recorded and displayed using

MATLAB 2019. To fit the typical evaluation protocol for unsupervised transfer learning tasks, the training datasets consist of 90% labelled data in the source domain and unlabelled data in the target domain, and the test datasets contain the remaining unlabelled data in the target domain.



N

RF

IF

OF

GPF

GBTF

Figure 7. Part of fault signals collected.

4.2. Experimental Results

To verify the significance of the proposed DTLCL approach, experiments were conducted with the same data set and other methods.

4.2.1. Comparison without Transfer with Individual Models

This study constructs seven different models including DTLCL, CNN, LSTM, AE, KNN, SVM and MMBT_mmbt [46], which is the Classic SOTA for classification. Table 3 displays each model's parameters. Both DTLCL and DNN are deep network structure models, which are composed of CNN and LSTM. Their structures and parameters are the same. The difference is that DTLCL first trains the DNN_s model with many significant fault samples, then transfers the trained model DNN_s to the micro-fault diagnosis model DNN_t, and finally trains the DNN_t model with a few micro-fault samples. A deep neural network model called DNN was trained using only a few micro-fault samples. The training iteration setting is 1000. The loss in CNN is cross-entropy. The AE encoding layer structure is 2048-128-64-6, and the decoding structure is just the opposite. All points in each domain of KNN have equal weights. The penalty parameter c in SVM is 1.1, its kernel is Gaussian kernel, and the degree parameter is 3. The dynamic learning rate (LR) is set as 0.01–0.0001 according to the epoch parameter during the training, LR is 0.01–0.0001. After a certain number of rounds, LR is gradually reduced. Near the completion of training, the learning rate declines by more than 100 times. In terms of Transfer Learning, since the model has converged on the original data set, the learning rate needs to be set as 0.0001.

Table 3. The hyper-parameters for models.

Model	Hyper-Parameters	Iterations
CNN	structure: (2048, 1) (128, 32) (64, 32) (2048) (100) (6)	1000
LSTM	structure: (2048, 1) (2048, 32) (65,536) (32) (32) (6)	1000
AE	encoder: 2048-128-32-6; decoder 6-32-128-2048	1000
KNN	n_neighbors: 5; $p = 1$; weight: uniform; leaf_size: 30	/
SVM	Cache size: 200, degree: 3 C:1.1	/
DTLCL	structure: (2048, 1) (128, 16) (64, 16) (64, 32) (32, 32) (32, 32) (16, 32) (16, 32) (512) (512) (32) (32) (6)	1000
DTLCL_A	structure: (2048, 1) (128, 16) (64, 16) (64, 32) (32, 32) (32, 32) (16, 32) (512) (512) (32) (32) (6)	1000
DTLCL_B	structure: (2048, 1)-(16, 32)-512-512-32-32-6	1000

The hyper-parameters can strongly influence the results. The parameter settings are usually divided into grid search, manual search and random search. This study proposes an improved method of grid search called step heap sorting. First, we set the initial and maximum values of the network parameters. Second, a fixed step is given to determine the next parameter to calculate the corresponding result of the parameter. Then the theoretical results are determined using heap sorting. Finally, the results are automatically compared, and the optimal parameters are determined, as mentioned in my other public paper.

To further verify that DTLCL combining CNN and LSTM produces better results than a single model-based transmission method, there is a simultaneous ablation experiment. The DTLCL model without LSTM is referred to as DTLCL_A based on the CNN-based transmission model without the LSTM structure, and the DTLCL model without CNN is referred to as DTLCL_B based on the LSTM-based transmission model without the CNN structure.

In Dataset A, the error size of 0.021 inches is large and the features are significant. In dataset B, the error with a magnitude of 0.0036 inches is small and the features are not obvious. The error with a magnitude of 0.021 inches is the source region and the error with a magnitude of 0.0036 inches is the target region in this experiment. This study aims to improve the diagnostic accuracy of errors with a size of 0.0036 inches by transfer learning.

To test the carryover effect from significant errors to smaller errors, two experiments are conducted in this section. Table 4 shows the conditions for significant and minor errors in Test 1 and Test 2. There are four types of significant errors, including N, IR, OR and RF. There are six types of minor errors, including N, IR, OR, RF, GPF, GBTF. The training dataset, verification dataset and significant error test dataset are divided into 1400, 400 and 100, respectively. The number of training datasets of micro-errors in Test 1 and Test 2 are different and are 50 and 200, respectively.

Table 4. Operating conditions of significant and minor faults in Test 1 and Test 2.

Working Condition	Significant Fault	Minor Fault
Load (hp)	1	2
Speed (rpm)	1750	1000
Fault size (inch)	0.021	0.0036
Status type	4 (NRF, IF, OF)	6 (N, RF, IF, OF, GPF, GBTF)

In Test 1, 2000 training samples are required for each severe fault, and 50 training samples are required for each minor fault.

In Test 2, for each major defect, there are 2000 training samples, and there are 200 training samples for each minor fault.

The results are shown in the table below. First, Table 5 displays the nine models' average training, validation, testing accuracy, and training time for Tests 1 and 2 for ten trials. It can be seen that DTLCL has the highest accuracy, 0.884, 0.876, 0.885 in Test 1 and 0.953, 0.946, 0.945 in Test 2, respectively, in the training, validation and test sets. The maximum average accuracy rates of the other models without transfer are 0.786, 0.731, 0.723 in test 1 and 0.886, 0.872, 0.877 in test 2. Moreover, the standard deviations of DTLCL in test 1 are 0.52, 0.43, 0.55 and the loss rates are 0.42, 0.43, 0.43. The standard deviations of DTLCL in test 2 are 0.32, 0.33, 0.34 and the loss rates are 0.38, 0.36, 0.37. The other models without transfer's smallest variances are 0.63, 0.72, 0.64 in Test 1 and 0.43, 0.41, 0.43 in Test 2. The training time for DTLCL is 18 s in Test 1 and 19 s in Test 2, while the minimum training time for CNN in other models without transfer costs 22 s in Test 1 and 23 s in Test 2. First, DTLCL has the characteristics of high accuracy, short detection time and low deviation. Second, the average error diagnosis accuracy of each model in Test 2 is higher than that of the corresponding model in Test 1, which demonstrates that the efficiency of DCLCL is proportional to the number of samples. The result shows that the accuracy and robustness of fault diagnosis by the proposed DTLCL are significantly improved by transfer learning and the number of samples.

Table 5. Average results over ten trials for six compared models.

Test	Model	Accuracy of Training/Loss	Accuracy of Validation/Loss	Accuracy of Testing/Loss	Time of Training
Test1	DTLCL	(0.884 ± 0.52)/0.42	(0.876 ± 0.43)/0.43	(0.885 ± 0.55)/0.43	18 s
	CNN	(0.786 ± 0.63)/0.64	(0.731 ± 0.72)/1.66	(0.723 ± 0.64)/1.78	22 s
	LSTM	(0767 ± 0.51)/1.52	(0.728 ± 0.62)/1.54	(0.707 ± 0.62)/1.53	26 s
	DNN	(0.602 ± 3.52)/1.57	(0.634 ± 3.52)/1.64	(0.645 ± 3.54)/1.47	35 s
	AE	(0.540 ± 3.52)/0.75	(0.562 ± 3.87)/1.44	(0.524 ± 3.72)/1.23	24 s
	KNN	(0.552 ± 1.22)	(0.541 ± 1.29)	(0.506 ± 1.32)	107 s
	SVM	(0.679 ± 3.82)	(0.651 ± 3.38)	(0.661 ± 3.38)	19 s
	DTLCL _A	(0.802 ± 0.41)/0.61	(0.798 ± 0.51)/0.58	(0.805 ± 0.45)/0.54	17 s
	DTLCL _B	(0.791 ± 0.58)/0.52	(0.787 ± 0.64)/0.53	(0.784 ± 0.55)/0.57	16 s
	MMBT_mmbt	(0.754 ± 0.51)/1.43	(0.713 ± 0.63)/1.48	(0.718 ± 0.60)/1.52	25 s
Test2	DTLCL	(0.953 ± 0.32)/0.38	(0.946 ± 0.33)/0.36	(0.945 ± 0.34)/0.37	19 s
	CNN	(0.886 ± 0.43)/0.58	(0.872 ± 0.41)/1.32	(0.877 ± 0.43)/1.22	23 s
	LSTM	(0854 ± 0.51)/0.62	(0.824 ± 0.52)/1.48	(0.835 ± 0.54)/1.37	30 s
	DNN	(0.872 ± 2.50)/1.23	(0.874 ± 2.54)/1.35	(0.885 ± 2.51)/1.33	39 s
	AE	(0.630 ± 3.42)/0.52	(0.664 ± 3.37)/1.12	(0.654 ± 3.22)/1.01	27 s
	KNN	(0.652 ± 1.01)	(0.641 ± 1.12)	(0.616 ± 1.28)	118 s
	SVM	(0.859 ± 3.82)	(0.840 ± 3.38)	(0.841 ± 3.38)	22 s
	DTLCL _A	(0.892 ± 0.41)/0.52	(0.898 ± 0.51)/0.53	(0.895 ± 0.45)/0.51	18 s
	DTLCL _B	(0.863 ± 0.58)/0.54	(0.868 ± 0.64)/0.56	(0.887 ± 0.55)/0.53	17 s
	MMBT_mmbt	(0.845 ± 0.54)/0.57	(0.831 ± 0.52)/1.38	(0.851 ± 0.53)/1.36	25 s

Note: The results are formatted as average (accuracy ± standard deviation)/error rate.

The particular test accuracy of the different methods in the experiments is shown in Figure 8. Two points can be intuitively deduced from the analysis of the results in Figure 8. First, DTLCL has the highest accuracy of the test data set in each experiment, about 80% in Test 1 and more than 95% in Test 2. Second, the accuracy of the DTLCL test data set is stable in each experiment. In contrast, alternative approaches without transfer produce weak, unstable, and less reliable outcomes. The average accuracy of the test data set of each model in Test 2 is higher than that of the corresponding model in Test 1. These results also show that the DTLCL method can be more accurate and stable than the other five models and is determined by the number of samples. The accuracy of the test dataset increases with sample size.

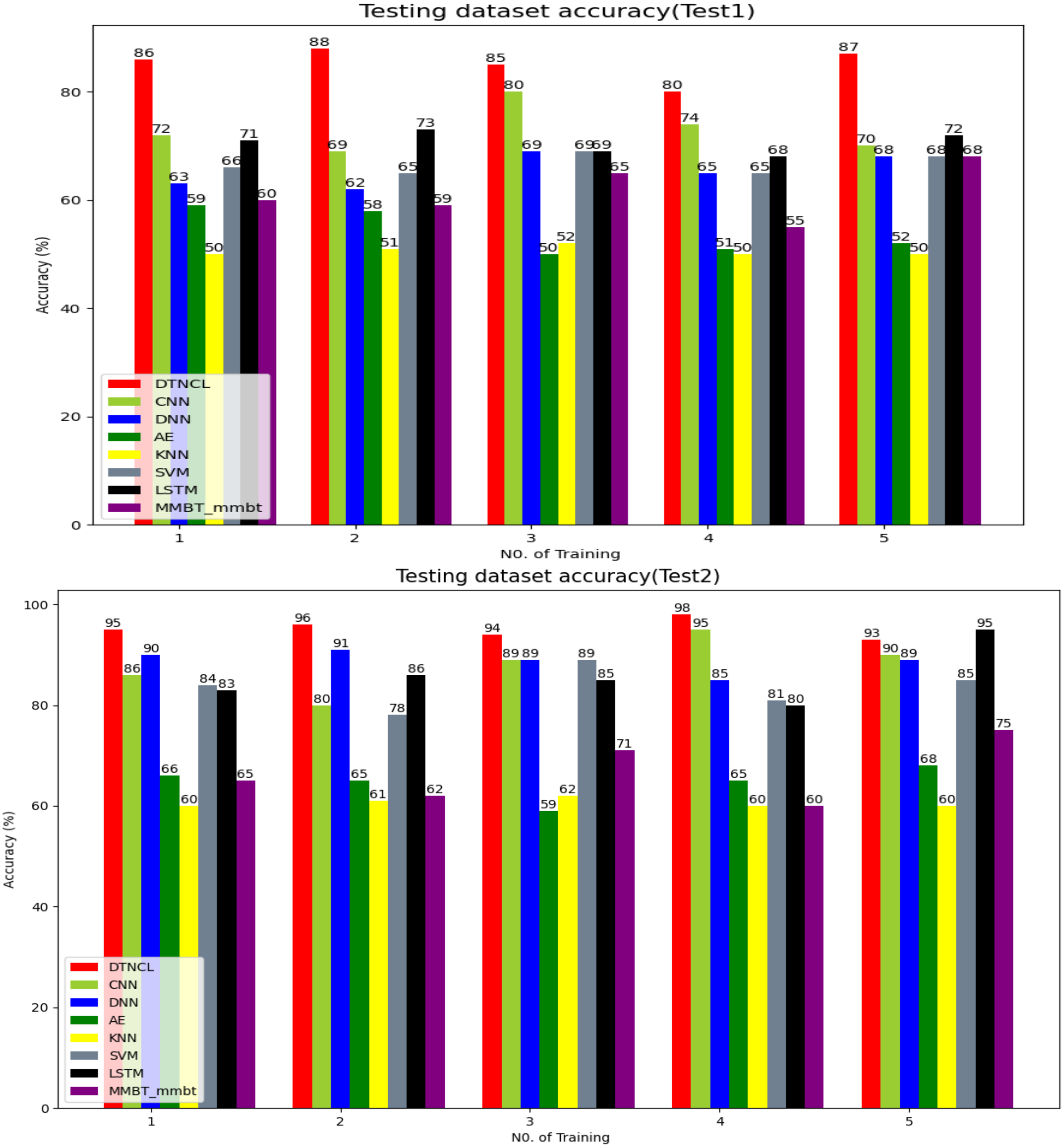


Figure 8. Comparison of accuracy of different methods in five experiments.

More focused studies have been conducted to further confirm the efficacy of the proposed DTLCL. In this section, the P, R and F values of several models are presented. Figure 9 shows the P-values of DTLCL and the other five models in the test data set for experiments. The P accuracy of DTLCL in Test 1 and Test 2 is the highest, especially in N, IF, RF, OF. The corresponding P-values of the other models are less than 60% in Test 1 and 70% in Test 2. Less than 50% in Test 1 and 65% in Test 2 show that the P-rate of KNN is extremely low at six errors. On the other hand, the precision rate is consistent at six errors and the value of the precision rate of the DTLCL model has increased to more than 80% in Test 1 and 95% in Test 2.

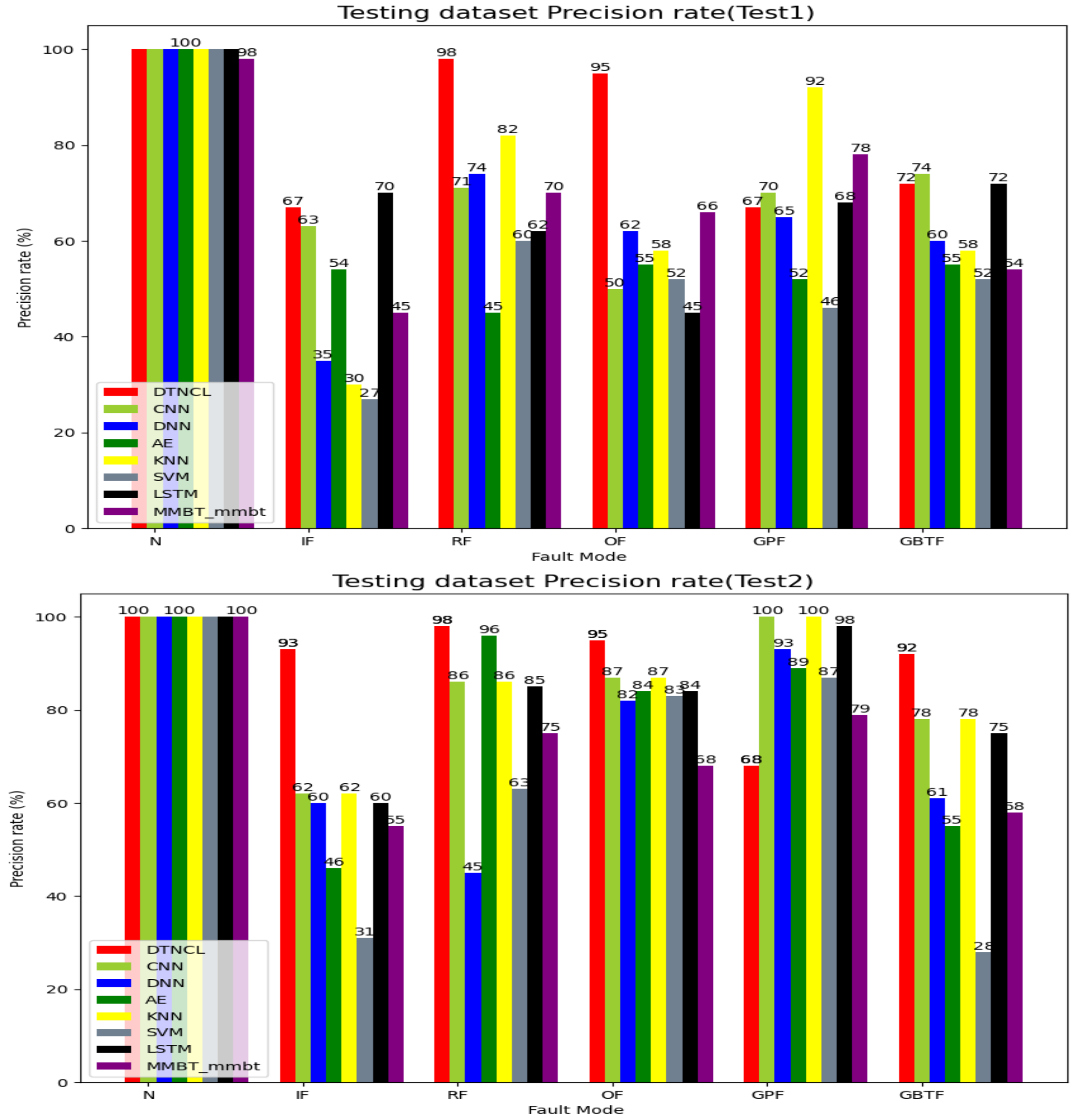


Figure 9. P of different models.

Figure 10 shows the recognition rate of DTLCL and six other models in the test data set. The results show that DTLCL has a higher recognition rate than the other models, especially for N, RF, GPF in Test 1 and N, IF, RF, OF, GPF in Test 2. The average recognition rate of the other models is less than 45% in Test 1 and 90% in Test 2. In contrast, the average recognition rate of the DTLCL models increases to more than 75% in Test 1 and 92% in Test 2.

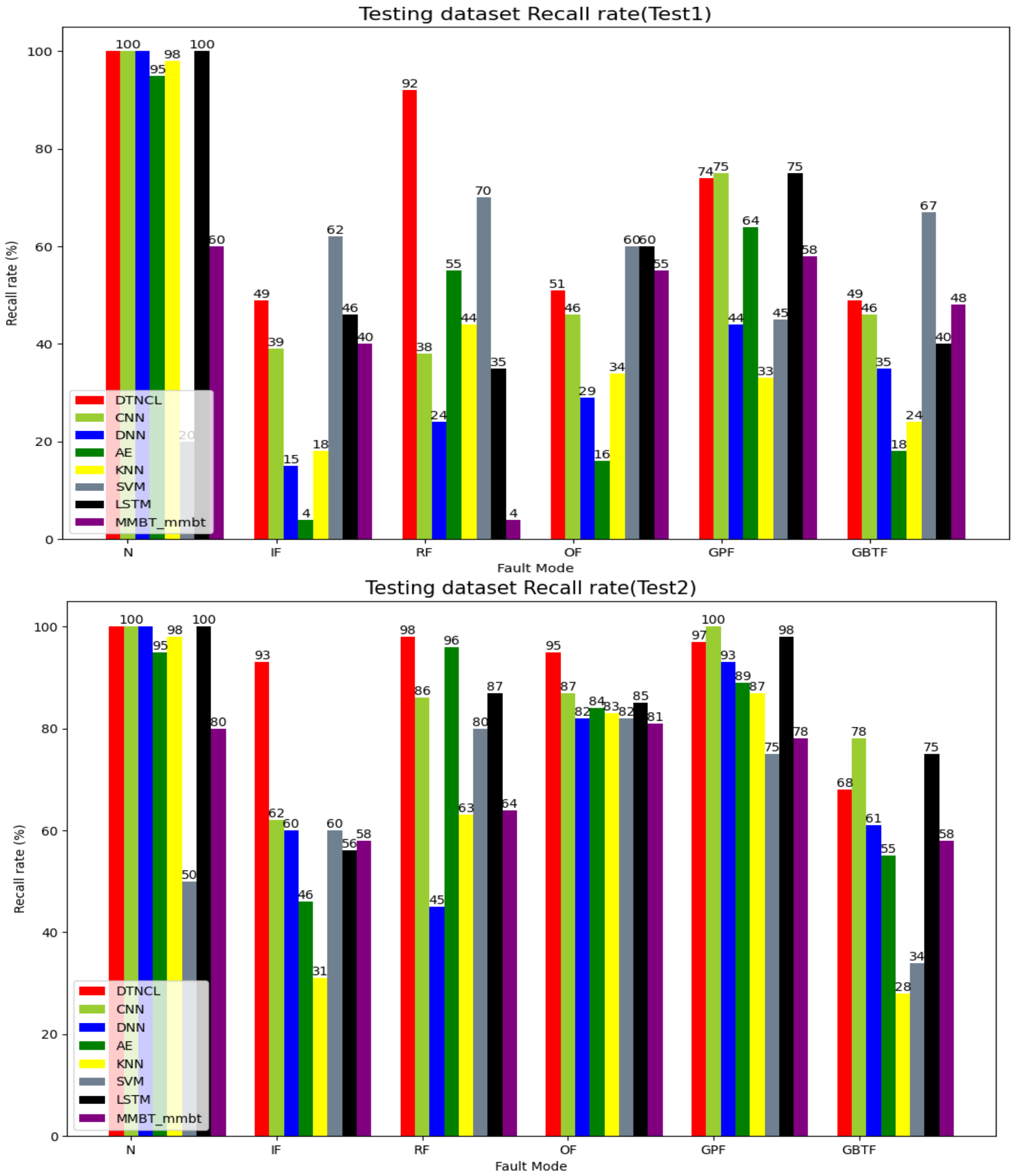


Figure 10. R of different models.

Precision and recall are well developed in DTLCL, but they cannot comprehensively and objectively evaluate the results of the models. F is a good index to evaluate models comprehensively. Figure 11 shows the test data set F of DTLCL and other different models. The F value of the DTL model is greater than 75% for every error, especially for N, OF, IF, RF, GPF in Test 1 and Test 2. Most of the other models are below 70%. The above results also confirm that compared to the other models without transmission, the DTLCL approach performs better.

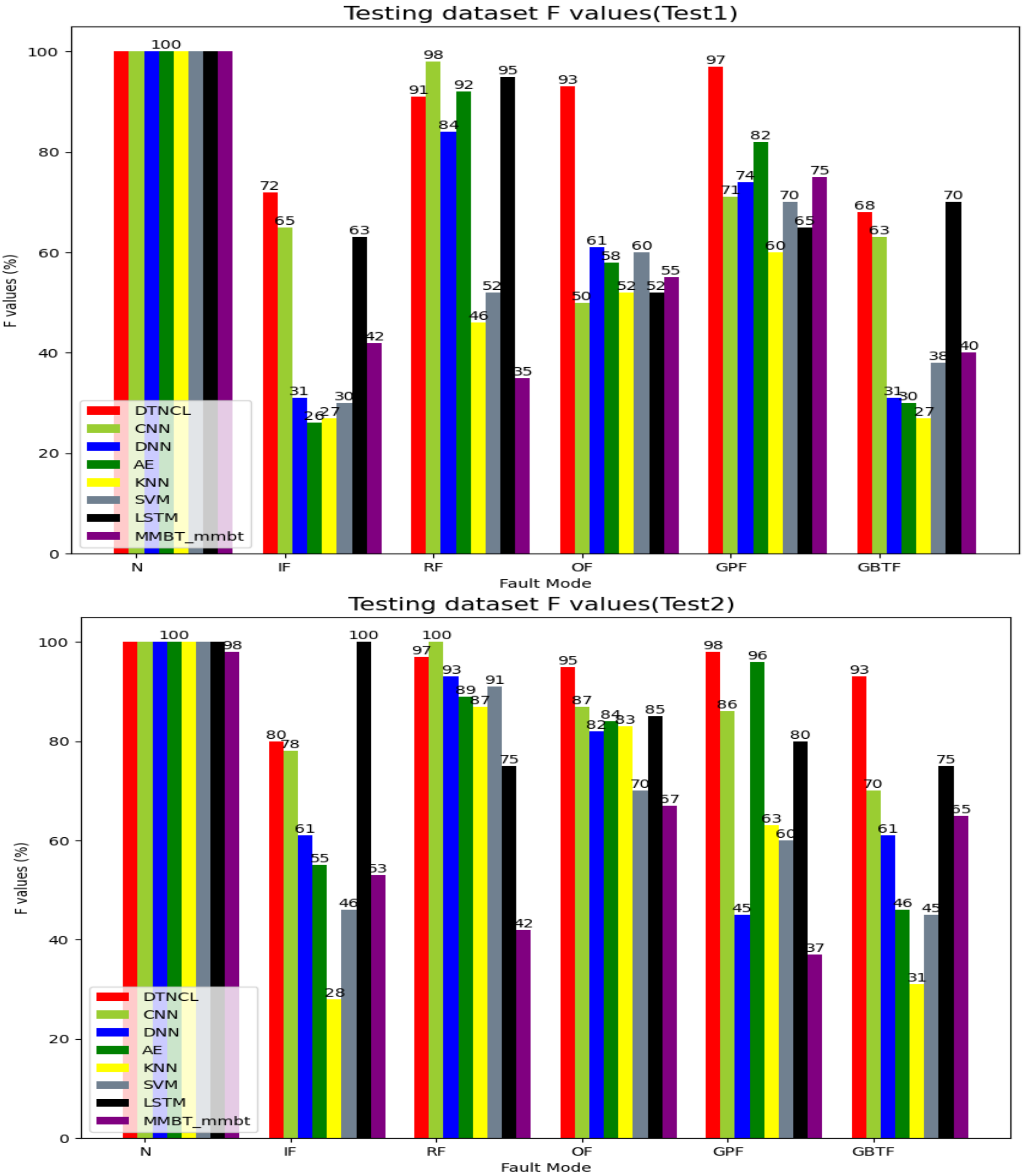


Figure 11. F of different models.

Based on the above results, it can be assumed that the indicators of accuracy, precision, detection and comprehensive assessment are improved by DTLCL. Moreover, the results of DTLCL are more precise, consistent, and generalizable depending on the number of samples. From the analysis of the above results, it can be inferred that DTLCL has improved the indicators of accuracy, precision, detection and comprehensive assessment. Moreover, depending on the number of samples, the results of DTLCL are more repeatable, accurate and have high generalisation ability. The accuracy rises with the size of the sample.

4.2.2. Comparison with Other Models with Transfer

To further demonstrate the transmission performance of the DTLCL model and explain the reason why this method is better than other fault diagnosis methods under different working conditions and faults of different degrees, the above training models of CNN, AE as TCNN, TAE are retained. The well-known deep models such as Xception, InceptionV3, D- CORAL, DANN can also be transferred. A model for transfer learning in SOTA models called DistilBERT_transformers is also used in comparative experiments [47]. Thus, TCNN, TAE, Xception, InceptionV3 are presented for comparison on dataset A and dataset B. A single Gaussian kernel is used on TCNN, TAE, TANN, TSVM. On DTLCL, a multi-tier adapter with a Gaussian combination of a linear kernel is used. D-CORAL and DNN share the same architecture but adapt to the properties of fully connected layer 1 and fully connected layer 2 by using CORAL loss. DANN has the same structure as DNN for fault classification but adds a domain classifier for domain adaptation that inputs flat layer features into the domain classifier.

In Test 3 and Test 4, the number of training samples is the same as in Test 1 with significant and small errors. Table 6 shows the accuracy and training time of the different models in the training dataset, validation dataset and test dataset among ten experiments. The following two effects can be intuitively deduced from the analysis of the results in Tables 5 and 6. First, the overall accuracy of the models with transfer is better than those without transfer and the training time is shortened. The average accuracy rates of the test data set TCNN, TAE, InceptionV3 and Xception with transfer are 0.843, 0.524, 0.630, 0.812 in Test 1 and 0.934, 0.752, 0.715, 0.864 in Test 2, respectively, the time taken is 19 s, 22 s, 80 s, 17 s and 20 s, 24 s, 81 s, 17 s. This shows that DTLCL with a Gaussian combination of linear kernels reduces the distribution discrepancy and gives better results.

Table 6. Average results over ten trials for five compared models with transfer.

Test	Model	Accuracy of Training	Accuracy of Validation	Accuracy of Testing	Time of Training
Test 3	DTLCL	0.884	0.876	0.885	18 s
	D-CORAL	0.834	0.823	0.837	21 s
	DANN	0.821	0.818	0.824	22 s
	TCNN	0.882	0.841	0.843	19 s
	TAE	0.580	0.562	0.524	22 s
	InceptionV3	0.612	0.651	0.557	27 s
	Xception	0.729	0.749	0.732	17 s
	DistilBERT_transformers	0.581	0.579	0.543	23 s
Test 4	DTLCL	0.953	0.946	0.945	19 s
	D-CORAL	0.926	0.915	0.933	22 s
	DANN	0.911	0.918	0.909	23 s
	TCNN	0.956	0.923	0.934	20 s
	TAE	0.734	0.781	0.752	24 s
	InceptionV3	0.754	0.702	0.715	28 s
	Xception	0.871	0.865	0.864	17 s
	DistilBERT_transformers	0.712	0.705	0.701	20 s

F allows for a more comprehensive evaluation of the models by taking both P and C into account. Figure 12 shows the specific F value of the different transfer models in ten tests. Except for GBTF, the F-value of DTLCL in Test 3 and Test 4 is higher than that of the other models, exceeding 75%. In particular, for error modes such as OF, RF, IF, the F-values of DTLCL improve significantly to 75.5% from about 55% in Test 3 and 94.5% from about 76% in Test 4 of the other models. These results show that the proposed DTLCL generally achieves higher F-values than other models.

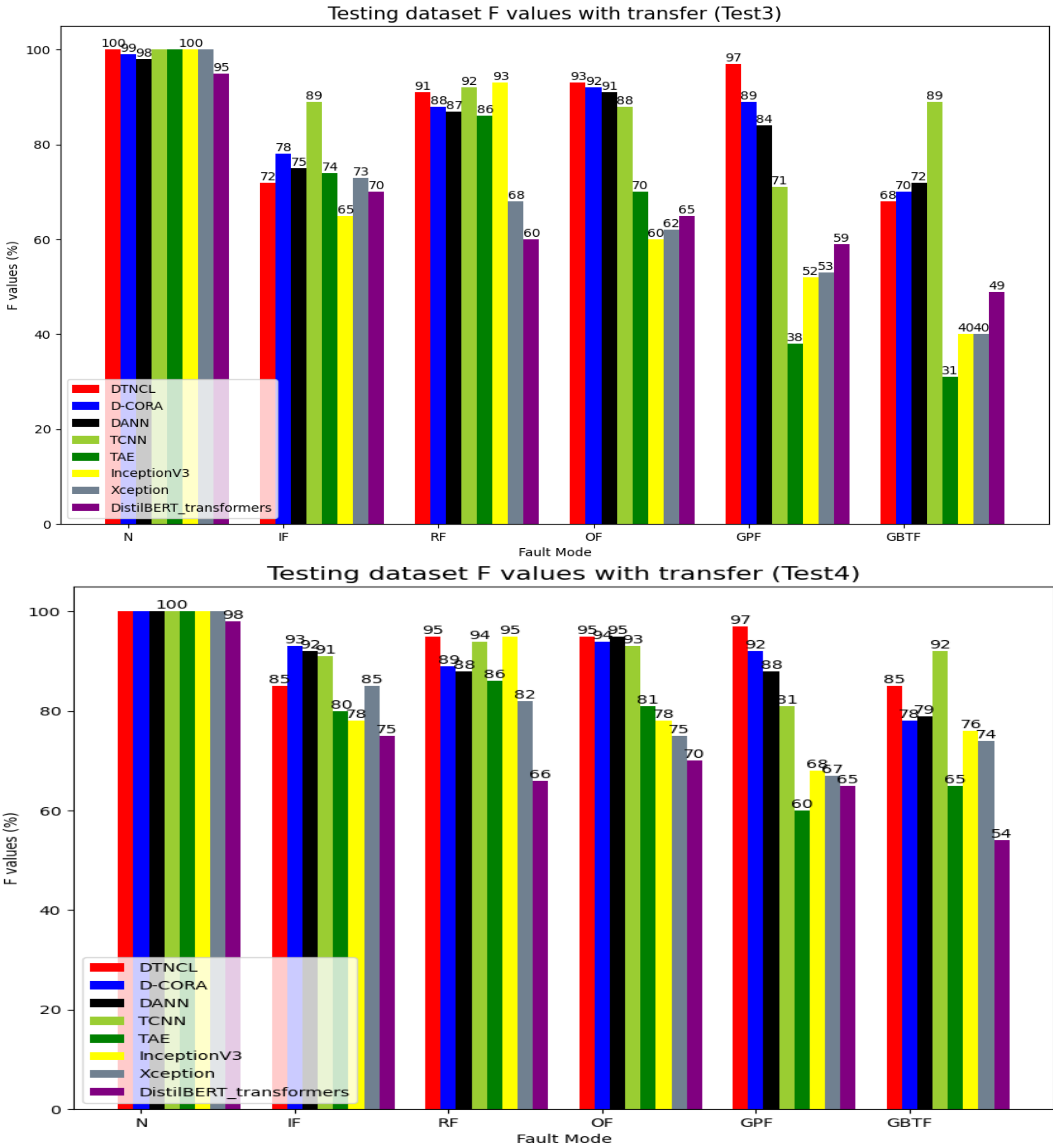


Figure 12. F of different transfer models.

4.3. Experiment Analysis

In this section, the proposed DTLCL approach is compared with alternative learning methods using different measurement techniques. From the above results, the following points can be summarised.

(1) DTLCL combines three components: Deep Learning, Model-Based Transfer Learning and Domain Adaptation. More importantly, DTLCL fully utilises each component to make it a system. Deep learning models can effectively extract features, and model-based transfer learning can effectively initialise the DTN built from two different kernel MMDs to achieve domain adaptation and generate transitive features. One-dimensional Convolu-

tional Neural Networks can learn features that are domain invariant due to the domain matching, which uses MMD to reduce the maximum mean among the source domain and the target domain; (2) In general, the DTLCL model exploits advantages such as deep learning, transfer learning and domain adaptation, outperforming other non-transfer and transfer models; (3) Compared to non-transfer models and transfer models, DTLCL does not require professional manual feature extraction to improve diagnostic results. It reflects the advantages of unsupervised deep transfer learning; and (4) DTLCL achieves much higher diagnostic accuracy than other models in Test 1 than in Test 2. This shows that the test dataset's accuracy increases with sample size.

The above conclusions show that DTLCL can perform better on unlabelled samples or micro-samples from the target area under different working conditions and error levels. The DTLCL model of the non-obvious micro-defect samples will be optimised to improve the accuracy of micro-defect diagnosis.

5. Conclusions

In this paper, the performance of rolling bearing micro-fault diagnosis for wireless charger production under different operating situations and fault levels is improved by a data-driven approach called DTLCL. The proposed approach makes use of transfer learning and deep learning. To improve the feature extraction capability, multi-kernel MMD is applied between the source and target domains. The effectiveness of the method was tested using dataset A and the actual measurement data of the warehouse (dataset B). The training time is 19 s and the accuracy exceeds 94.5%. The explanation results show that the proposed DTLCL method provides higher accuracy and robust identification of smaller errors compared to the current combination of integrated combinations and single non-transmission models. Thus, the DTLCL method could be used for fault diagnosis of bearings and gears, further promoting the application process of wireless charging. At the same time, the WV method can accurately and quickly determine the hyperparameters of the model, improving the accuracy of the model. As bearings and gears are among the most critical components in the manufacturing of wireless charging devices, the developed method can be used to identify the associated micro-defects, which improves the functionality of wireless charging applications.

Author Contributions: Conceptualization, W.L.; Data curation, Y.W.; Formal analysis, Y.W.; Funding acquisition, W.L.; Investigation, Y.W.; Methodology, Y.W. and H.Z.; Project administration, W.L.; Software, Y.W.; Supervision, W.L.; Validation, Y.W.; Writing—original draft, Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: Supported by the National Natural Science Foundation of China and Project No. 51975444.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is unavailable due to privacy or ethical restrictions.

Acknowledgments: The author would like to thank Li Weidong, Qiao Peng and colleagues from the university where the author works for their review comments.

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

Nomenclature

DL	deep learning
SAE	stacked auto-encoder
DTLCL	a transfer-based deep neural network
MTL	model-based transfer learning
DNN	deep neural network
DTN	deep transfer network
MMD	maximum mean discrepancy

SAE	stacked auto-encoder
CNN	convolutional neural network
DBN	deep belief network
RNN	recurrent neural network
LSTM	long short-term memory
ML	machine learning
SVM	support vector machine
LR	logistic regression
k-NN	k-nearest neighbour
DDA	deep domain adaptation
WV	weighted voting
TL	transfer learning
ITL	instance-based transfer learning
FTL	feature-based transfer learning
MTL	model-based transfer learning
RTL	relation-based transfer learning
DNN	deep neural network
DDA	deep domain adaptation
Ds	source domain
Dt	target domain
S	source domain dataset
N_s	the total number of samples
y	actual label
S_i	the *i*th sample
p	the dimensionality of the sample
T	target domain dataset without label
N_t	the total number of the sample
T_j	the *j*th sample
k	the kernel
DNNs	the model of significant faults
DNNt	the model of minor faults
X_s	the training dataset with many significant fault samples
X_i	the training dataset with only a few minor fault samples
$\mathbf{x}^*$	min–max mormalization
x	the pre-conversion value
x^*	the converted value
max	the maximum value of the original data
min	the minimum value of the original data
SMOTE	synthetic minority oversampling technique
f_{sj}	the number of neurons in the jth hidden layer of the DNN_s
X_s	the training dataset from significant fault.
θ_s	the initial set of parameters for the network DNN_s
θ_{si}	the set of parameters of the weight matrix and bias of the input layer
$\theta_{s'}$.	layer-by-layer training updates DNN_s parameters
F_{sN}	abstract features
F_{cnn}	the comprehensive evaluation index for single CNN model training
F_{lstm}	the comprehensive evaluation index for single LSTM model training
TP	true positive
FP	false positive
FN	false negative
$\theta_{ss'}$	parameter of softmax from DNN_s
$\theta_{ts'}$	parameter of softmax from DNN_t
β	random initialization
P	precision
R	recall
F	comprehensive index

CWRU	Case Western Reserve University
N	normal fault
OF	outer circle fault
GPF	gear pitting fault
IF	inner circle fault
GBTF	gear broken tooth fault
RF	rolling element fault

References

1. Wang, G.; Zhao, G.; Xie, J.; Liu, K. Ensemble Learning Based Correlation Coefficient Method for Robust Diagnosis of Voltage Sensor and Short-Circuit Faults in Series Battery Packs. *IEEE Trans. Power Electron.* **2023**, *38*, 9143–9156. [CrossRef]
2. Shahapure, S.B.; Kulkarni, V.A.; Shinde, S.M. A Technology Review of Energy Storage Systems, Battery Charging Methods and Market Analysis of EV Based on Electric Drives. *Int. J. Electr. Electron. Res.* **2022**, *10*, 23–35. [CrossRef]
3. Dong, Y.; Lu, W.; Chen, H. Optimization Study for Lateral Offset Tolerance of Electric Vehicles Dynamic Wireless Charging. *IEEJ Trans. Electr. Electron. Eng.* **2020**, *15*, 1219–1229. [CrossRef]
4. Feng, H.; Tavakoli, R.; Onar, O.C.; Pantic, Z. Advances in High-Power Wireless Charging Systems: Overview and Design Considerations. *IEEE Trans. Transp. Electrif.* **2020**, *6*, 886–919. [CrossRef]
5. Jia, F.; Lei, Y.; Lin, J.; Zhou, X.; Lu, N. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mech. Syst. Signal Process.* **2016**, *72–73*, 303–315. [CrossRef]
6. Shao, H.; Jiang, H.; Wang, F.; Zhao, H. An enhancement deep feature fusion method for rotating machinery fault diagnosis. *Knowl.-Based Syst.* **2017**, *119*, 200–220. [CrossRef]
7. Chen, Z.; Deng, S.; Chen, X.; Li, C.; Sanchez, R.-V.; Qin, H. Deep neural networks-based rolling bearing fault diagnosis. *Microelectron. Reliab.* **2017**, *75*, 327–333. [CrossRef]
8. Li, C.; Sánchez, R.-V.; Zurita, G.; Cerrada, M.; Cabrera, D. Fault Diagnosis for Rotating Machinery Using Vibration Measurement Deep Statistical Feature Learning. *Sensors* **2016**, *16*, 895. [CrossRef]
9. Shao, H.; Jiang, H.; Zhang, X.; Niu, M. Rolling bearing fault diagnosis using an optimization deep belief network. *Meas. Sci. Technol.* **2015**, *26*, 115002. [CrossRef]
10. Janssens, O.; Slavkovikj, V.; Vervisch, B.; Stockman, K.; Loccufier, M.; Verstockt, S.; Van de Walle, R.; Van Hoecke, S. Convolutional Neural Network Based Fault Detection for Rotating Machinery. *J. Sound Vib.* **2016**, *377*, 331–345. [CrossRef]
11. Guo, X.; Chen, L.; Shen, C. Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis. *Measurement* **2016**, *93*, 490–502. [CrossRef]
12. Hu, Y.; Wang, Y.; Wang, H. A decoding method based on RNN for OvTDM. *China Commun.* **2020**, *17*, 1–10. [CrossRef]
13. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A Comprehensive Survey on Transfer Learning. *Proc. IEEE* **2021**, *109*, 43–76. [CrossRef]
14. Bi, J.; Lee, J.-C.; Liu, H. Performance Comparison of Long Short-Term Memory and a Temporal Convolutional Network for State of Health Estimation of a Lithium-Ion Battery Using Its Charging Characteristics. *Energies* **2022**, *15*, 2448. [CrossRef]
15. Kathigi, A.; Krishnappa, H.K. Handwritten Character Recognition Using Unsupervised Feature Selection and Multi Support Vector Machine Classifier. *Int. J. Intell. Eng. Syst.* **2021**, *14*, 290–300. [CrossRef]
16. Lee, J.M. Fast k-Nearest Neighbor Searching in Static Objects. *Wirel. Pers. Commun.* **2017**, *93*, 147–160. [CrossRef]
17. Liu, K.; Peng, Q.; Teodorescu, R.; Foley, A.M. Knowledge-Guided Data-Driven Model with Transfer Concept for Battery Calendar Ageing Trajectory Prediction. *IEEE/CAA J. Autom. Sin.* **2023**, *10*, 272–274. [CrossRef]
18. Liu, K.; Peng, Q.; Che, Y.; Zheng, Y.; Li, K.; Teodorescu, R.; Widanage, D.; Barai, A. Transfer learning for battery smarter state estimation and ageing prognostics: Recent progress, challenges, and prospects. *Adv. Appl. Energy* **2023**, *9*, 100117. [CrossRef]
19. Choudhry, A.; Khatri, I.; Jain, M.; Vishwakarma, D.K. An Emotion-Aware Multitask Approach to Fake News and Rumour Detection using Transfer Learning. *IEEE Trans. Comput. Soc. Syst.* 2022, *early access*.
20. Peng, Q.; Liu, W.; Zhang, Y.; Zeng, S.; Graham, B. Generation planning for power companies with hybrid production technologies under multiple renewable energy policies. *Renew. Sustain. Energy Rev.* **2023**, *176*, 113209. [CrossRef]
21. Zhu, Y.; Zhuang, F.; Wang, J.; Chen, J.; Shi, Z.; Wu, W.; He, Q. Multi-representation adaptation network for cross-domain image classification. *Neural Netw.* **2019**, *119*, 214–221. [CrossRef]
22. Lin, W.; Mak, M.-W.; Li, L.; Chien, J.-T. Reducing Domain Mismatch by Maximum Mean Discrepancy Based Autoencoders. In Proceedings of the Speaker and Language Recognition Workshop (Odyssey 2018), Les Sables d'Olonne, France, 26–29 June 2018. [CrossRef]
23. Sapkota, U.; Solorio, T.; Montes, M.; Bethard, S. Domain Adaptation for Authorship Attribution: Improved Structural Correspondence Learning. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016. [CrossRef]
24. Sanodiya, R.K.; Mishra, S.; Arun, P.V. Manifold embedded joint geometrical and statistical alignment for visual domain adaptation. *Knowl.-Based Syst.* **2022**, *257*, 109886. [CrossRef]
25. Li, Y.; Zheng, H.; Zhu, H.; Ai, H.; Dong, X. Cross-People Mobile-Phone Based Airwriting Character Recognition. In Proceedings of the 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021. [CrossRef]

26. Davis, J.; Domingos, P. Deep Transfer: A Markov Logic Approach. *AI Mag.* **2011**, *32*, 51–53. [CrossRef]
27. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-Adversarial Training of Neural Networks. In *Domain Adaptation in Computer Vision Applications*; Advances in Computer Vision and Pattern Recognition; Springer: Cham, Switzerland, 2017; pp. 189–209. [CrossRef]
28. Bousmalis, K.; Trigeorgis, G.; Silberman, N.; Krishnan, D.; Erhan, D. Domain Separation Networks. *arXiv* **2016**, arXiv:1608.06019.
29. Chen, D.; Yang, S.; Zhou, F. Transfer Learning Based Fault Diagnosis with Missing Data Due to Multi-Rate Sampling. *Sensors* **2019**, *19*, 1826. [CrossRef] [PubMed]
30. Han, T.; Liu, C.; Yang, W.; Jiang, D. Learning transferable features in deep convolutional neural networks for diagnosing unseen machine conditions. *ISA Trans.* **2019**, *93*, 341–353. [CrossRef]
31. Shukla, V.; Choudhary, S. Deep Learning in Neural Networks: An Overview. In *Deep Learning in Visual Computing and Signal Processing*; Apple Academic Press: Palm Bay, FL, USA, 2022; pp. 29–53. [CrossRef]
32. Long, M.; Wang, J.; Cao, Y.; Sun, J.; Yu, P.S. Deep Learning of Transferable Representation for Scalable Domain Adaptation. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 2027–2040. [CrossRef]
33. Jwa, H.; Oh, D.; Park, K.; Kang, J.; Lim, H. exBAKE: Automatic Fake News Detection Model Based on Bidirectional Encoder Representations from Transformers (BERT). *Appl. Sci.* **2019**, *9*, 4062. [CrossRef]
34. Sahoo, S.R.; Gupta, B.B. Multiple features based approach for automatic fake news detection on social networks using deep learning. *Appl. Soft Comput.* **2021**, *100*, 106983. [CrossRef]
35. Azamfar, M.; Li, X.; Lee, J. Intelligent ball screw fault diagnosis using a deep domain adaptation methodology. *Mech. Mach. Theory* **2020**, *151*, 103932. [CrossRef]
36. Guo, L.; Lei, Y.; Xing, S.; Yan, T.; Li, N. Deep Convolutional Transfer Learning Network: A New Method for Intelligent Fault Diagnosis of Machines with Unlabeled Data. *IEEE Trans. Ind. Electron.* **2019**, *66*, 7316–7325. [CrossRef]
37. Xie, Y.; Zhao, J.; Qiang, B.; Mi, L.; Tang, C.; Li, L. Attention Mechanism-Based CNN-LSTM Model for Wind Turbine Fault Prediction Using SSN Ontology Annotation. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 6627588. [CrossRef]
38. He, Z.; Shao, H.; Zhang, X.; Cheng, J.; Yang, Y. Improved Deep Transfer Auto-Encoder for Fault Diagnosis of Gearbox Under Variable Working Conditions with Small Training Samples. *IEEE Access* **2019**, *7*, 115368–115377. [CrossRef]
39. Hemalatha, P.; Amalanathan, G.M. FG-SMOTE: Fuzzy-based Gaussian synthetic minority oversampling with deep belief networks classifier for skewed class distribution. *Int. J. Intell. Comput. Cybern.* **2021**, *14*, 270–287. [CrossRef]
40. Li, J.; Zhu, Q.; Wu, Q.; Fan, Z. A novel oversampling technique for class-imbalanced learning based on SMOTE and natural neighbors. *Inf. Sci.* **2021**, *565*, 438–455. [CrossRef]
41. Chakraborty, D.; Narayanan, V.; Ghosh, A. Integration of deep feature extraction and ensemble learning for outlier detection. *Pattern Recognit.* **2019**, *89*, 161–171. [CrossRef]
42. Ma, S.; Chu, F. Ensemble deep learning-based fault diagnosis of rotor bearing systems. *Comput. Ind.* **2019**, *105*, 143–152. [CrossRef]
43. Wang, Z.-Y.; Lu, C.; Zhou, B. Fault diagnosis for rotary machinery with selective ensemble neural networks. *Mech. Syst. Signal Process.* **2018**, *113*, 112–130. [CrossRef]
44. Shao, H.; Jiang, H.; Lin, Y.; Li, X. A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders. *Mech. Syst. Signal Process.* **2018**, *102*, 278–297. [CrossRef]
45. Case Western Reserve University, Bearing Data Center. Available online: https://engineering.case.edu/bearingdatacenter/download-data-file (accessed on 23 September 2020).
46. Machine Heart Technology Co., Ltd. Available online: https://sota.jiqizhixin.com/implements/facebookresearch-mmbt_mmbt (accessed on 1 June 2021).
47. Machine Heart Technology Co., Ltd. Available online: https://sota.jiqizhixin.com/implements/huggingface-distilbert_transformers (accessed on 6 August 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.