

## Article

# A RUL Prediction Method of Small Sample Equipment Based on DCNN-BiLSTM and Domain Adaptation

Wenbai Chen <sup>1,\*</sup>, Weizhao Chen <sup>1</sup>, Huixiang Liu <sup>1</sup>, Yiqun Wang <sup>1</sup>, Chunli Bi <sup>2</sup> and Yu Gu <sup>3,4,5</sup>

<sup>1</sup> School of Automation, Beijing Information Science and Technology University, Beijing 100101, China; chenwz312@163.com (W.C.); liuhx@bistu.edu.cn (H.L.); wangyiqun@bistu.edu.cn (Y.W.)

<sup>2</sup> China Academy of Information and Communications Technology, Beijing 100191, China; bichunli@caict.ac.cn

<sup>3</sup> Guangdong Province Key Laboratory of Petrochemical Equipment Fault Diagnosis, Guangdong University of Petrochemical Technology, Maoming 525000, China; guyufrankfurt@163.com

<sup>4</sup> College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China

<sup>5</sup> Department of Chemistry, Institute of Inorganic and Analytical Chemistry, Goethe-University, Max-von-Laue-Str. 9, 60438 Frankfurt, Germany

\* Correspondence: chenwb@bistu.edu.cn

**Abstract:** To solve the problem of low accuracy of remaining useful life (RUL) prediction caused by insufficient sample data of equipment under complex operating conditions, an RUL prediction method of small sample equipment based on a deep convolutional neural network—bidirectional long short-term memory network (DCNN-BiLSTM) and domain adaptation is proposed. Firstly, in order to extract the common features of the equipment under the condition of sufficient samples, a network model that combines the deep convolutional neural network (DCNN) and the bidirectional long short-term memory network (BiLSTM) was used to train the source domain and target domain data simultaneously. The Maximum Mean Discrepancy (MMD) was used to constrain the distribution difference and achieve adaptive matching and feature alignment between the target domain samples and the source domain samples. After obtaining the pre-trained model, fine-tuning was used to transfer the network structure and parameters of the pre-trained model to the target domain for training, perform network optimization and finally obtain an RUL prediction model that was more suitable for the target domain data. The method was validated on a simulation dataset of commercial modular aero-propulsion provided by NASA, and the experimental results show that the method improves the prediction accuracy and generalization ability of equipment RUL under cross-working conditions and small sample conditions.

**Keywords:** DCNN-BiLSTM; domain adaptation; MMD; fine-tuning; C-MAPSS; cross-working; small sample



**Citation:** Chen, W.; Chen, W.; Liu, H.; Wang, Y.; Bi, C.; Gu, Y. A RUL Prediction Method of Small Sample Equipment Based on DCNN-BiLSTM and Domain Adaptation. *Mathematics* **2022**, *10*, 1022. <https://doi.org/10.3390/math10071022>

Academic Editors: Jianping Gou, Weihua Ou, Shaoning Zeng and Lan Du

Received: 16 February 2022

Accepted: 19 March 2022

Published: 23 March 2022

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



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

## 1. Introduction

As one of the key technologies of Prognosis and Health Management (PHM), RUL prediction has become an important research content. RUL refers to the length of continuous working time of equipment components or systems from the current moment to the moment when a specific function cannot be performed [1]. Accurate RUL prediction plays a crucial role in guaranteeing system reliability and preventing system failures [2].

At present, the widely studied equipment RUL prediction methods can be divided into physical model-based methods and data-driven methods [3]. Due to the complex structure of some systems, the diverse failure modes, and the uncertainty of operating conditions, it is difficult to establish a physical failure model [4]. Data-driven methods without prior knowledge and complex physical modeling process [5] have become a research hotspot in recent years. Among them, deep learning has attracted much attention due to its powerful nonlinear mapping ability and high-dimensional feature extraction ability [6]. Babu et al. [7]

first tried to use the Convolutional Neural Network (CNN) to apply it to the RUL prediction of aero-engines. This model can automatically extract multi-dimensional sensor features and obtain better results than the shallow regression model. Zheng et al. [8] proposed a prediction model based on a Long Short-Term Memory (LSTM) network, which can extract the features of time series, is suitable for RUL prediction of most equipment.

The premise of data-driven methods is that the training and test data come from the same operating conditions. As a new machine learning method, transfer learning relaxes the premise that training samples and test samples must obey the same data distribution. The knowledge learned from the source domain is applied to different but related target domains to solve the problem of only a small number of labeled sample data in the target domain. Transfer learning improves the generalization ability of the machine learning model to a certain extent [9]. When the feature space and data distribution between the source domain and target domain samples are quite different, how to use the transfer learning strategy to solve the small sample problem becomes the focus of research.

Domain adaptation is an important research direction in transfer learning, which is used to solve the problem of transfer learning when the feature space and category space of two domains are consistent but the feature distribution is inconsistent. Domain adaptation methods have been used in the field of RUL prediction of equipment. Fu et al. [10] proposed a domain adaptation SAE-LSTM model, which adopted MMD to reduce the data distribution difference in RUL prediction. Li et al. [11] first proposed a multi-core MMD-based convolutional neural network model. Ragab [12] proposed a Contrastive Adversarial Domain Adaptation (CADA) method to learn similar features between different domains and improve the RUL prediction accuracy and noise immunity. Miao [13] proposed a Deep Domain Adaptive Network (DDAN) to solve the problem of cross-domain feature distribution shift under different operating conditions and failure modes. Costa et al. [14] proposed a domain adaptation method for RUL prediction under cross-working conditions based on LSTM and Domain Adversarial Neural Network (DANN). In order to solve the problem of low RUL prediction accuracy caused by small sample data sets, Lv et al. [15] proposed a Sequence Adaptation Adversarial Network (SAAN) to expand the dataset.

Traditional deep learning relies heavily on labeled data. Therefore, in view of the problem that small-sample equipment status data under different working conditions affect the RUL prediction accuracy, this paper proposes a small-sample equipment RUL prediction method based on DCNN-BiLSTM and domain adaptation. The model includes a pre-training stage, a parameter-transfer stage, and an RUL predicting stage. The pre-training and MMD constraints are used to reduce the distribution differences of sample data under different working conditions and learn the common characteristics of the source domain samples and the target domain samples after domain adaptation. Then transfer the trained model to the target domain for training fine-tune the pre-trained model to obtain an RUL prediction model more suitable for the target domain task. Finally, the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset provided by NASA was used to verify the effectiveness of the method proposed in this paper.

## 2. Related Work

### 2.1. CNN Convolution Model

The CNN has powerful parameter learning and feature extraction capabilities and can be used to process multi-dimensional matrix data. In practical engineering applications, each device has multiple sensors to detect the operating status of the device, and the collected data also contains a lot of information. In order to extract deeper features, this paper used a DCNN, which consists of multiple layers of CNN.

Since the degradation data of the equipment is the time series data collected by the sensor, in this study, the input data is a two-dimensional vector, the length represents the number of features collected by the sensor, and the width represents the time series of each feature. After the two-dimensional data is processed by time window, the size of each sample obtained is represented as  $(N_w, m)$ , where  $N_w$  represents the size of the time window

and  $m$  represents the number of features. Each convolutional layer performs convolution operations on the input data along the time series direction through convolution kernels of different sizes, which can extract different features between the data, and finally combine the generated local feature maps as the input of the BiLSTM.

2.2. BiLSTM Network Model

The LSTM model is used to process sequence data. Compared with the Recurrent Neural Network (RNN), LSTM is mainly used to solve the problems of gradient disappearance and gradient explosion in the training process of long sequence data. The LSTM model consists of an input layer, a hidden layer, and an output layer, with three gating units and memory units, and the historical information is affected by the input gate, forgetting gate, and output gate, respectively [16]. The dependencies between long and short periods of time series can be better learned.

As shown in Figure 1,  $i_t, o_t, f_t$  represent the input gate, output gate, and forget gate, respectively. The forget gate decides whether to retain the previous cell state information  $C_{t-1}$ ; the input gate updates the long-term memory of the cell state; the output gate is the output of the current LSTM;  $\tilde{C}_t$  represents the current temporary memory unit;  $x_t$  represents the time series of moments  $t$ ;  $h_{t-1}$  represents the output value of the previous moment;  $h_t$  represents the output value of the current moment. Then the calculation formula of each threshold state in the forward propagation process of LSTM is as follows:

$$\tilde{C}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{1}$$

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t \tag{2}$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{4}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) \tag{5}$$

$$h_t = o_t \cdot \tanh(c_t) \tag{6}$$

where  $\sigma$  represents the sigmoid activation function,  $\tanh$  is the hyperbolic tangent activation function,  $W_{xc}, W_{hc}, W_{xi}, W_{hi}, W_{ci}, W_{xf}, W_{hf}, W_{cf}, W_{xo}, W_{ho}, W_{co}, b_c, b_i, b_f,$  and  $b_o$  represent the weights and bias terms of each respective gate. LSTM contains many neurons, and the neurons exchange information with each other to extract time-dependent features of the data.

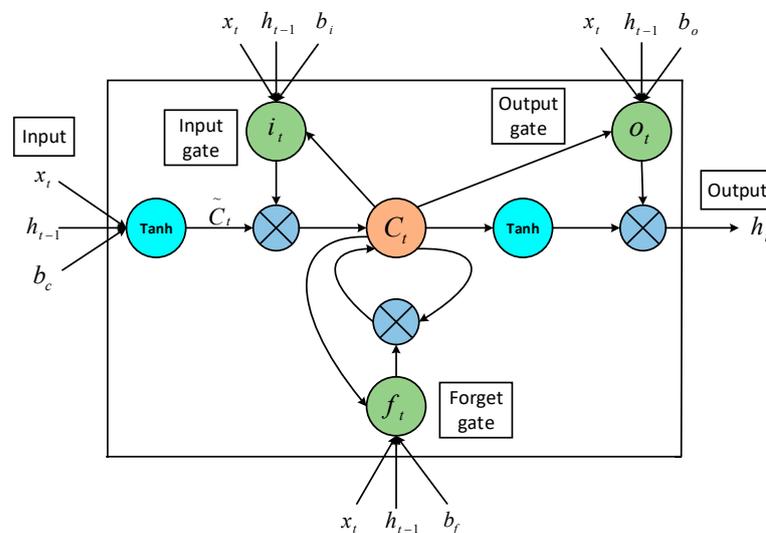


Figure 1. LSTM structure diagram.

Bi-directional LSTM (BiLSTM) contains two LSTM network layers in opposite directions, namely the forward propagation layer and the backward propagation layer, which connect the input layer and the output layer at the same time, perform time-sequence and reverse-order calculations, respectively, and obtain the output of the forward and backward hidden layer at each moment in turn. Finally, the final output is obtained by combining the corresponding output results of the forward layer and the backward layer at each moment. The BiLSTM structure diagram is shown in Figure 2. The specific calculation formula is as follows:

$$h_t = f(w_1x_t + w_2h_{t-1}) \tag{7}$$

$$h'_t = f(w_3x_t + w_5h'_{t-1}) \tag{8}$$

$$o_t = g(w_4x_t + w_6h'_t) \tag{9}$$

where  $h_t$  and  $h'_t$  are the outputs of the forward propagation layer and the backward propagation layer at time  $t$ , respectively.  $w_1$  and  $w_3$  are the weight matrices from the input layer to the forward and backward propagation layers, respectively.  $w_2$  and  $w_5$  are the weight matrices from the forward and backward propagation layers to the self-propagation layer, respectively.  $w_4$  and  $w_6$  are the weight matrices from the forward and backward propagation layers to the output layer, respectively.  $o_t$  is the output values of the final output gate.  $g$  are the functions for splicing the forward and backward propagation results.

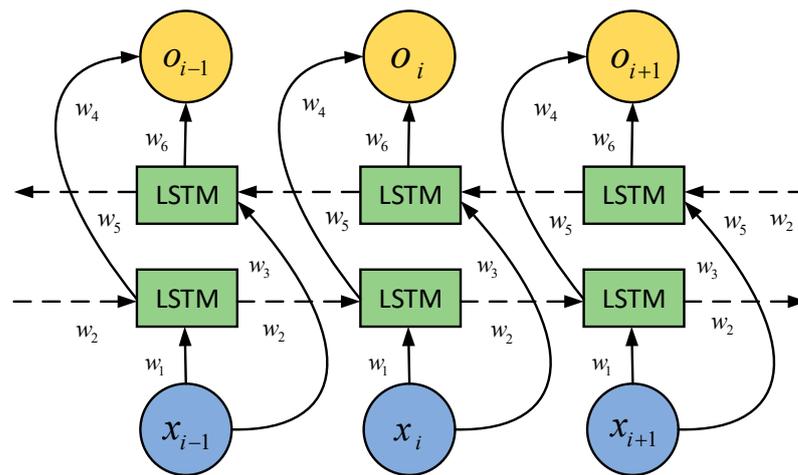


Figure 2. BiLSTM structure diagram.

### 3. Proposed Method

#### 3.1. RUL Prediction Model Based on DCNN-BiLSTM

The multi-dimensional sensor data obtained through time window processing is used as the input of the DCNN-BiLSTM fusion model, and the structure of the fusion model is shown in Figure 3. DCNN and BiLSTM process the input data, where DCNN consists of four layers of CNN and activation functions. Each layer of CNN performs low-level feature extraction by setting convolution kernels of different sizes, and then input to two layers of BiLSTM to extract time-series features, and finally two layers of fully connected layers. The BiLSTM network can comprehensively consider the historical information and future information at each moment and make full use of the information of the previous and subsequent moments to make the feature extraction process more comprehensive, improve the prediction accuracy of the time series model, and reduce the risk of overfitting. The output of the first fully connected layer is used as the measurement value of MMD. The second layer is the final prediction layer, and the output represents the RUL value of the device.

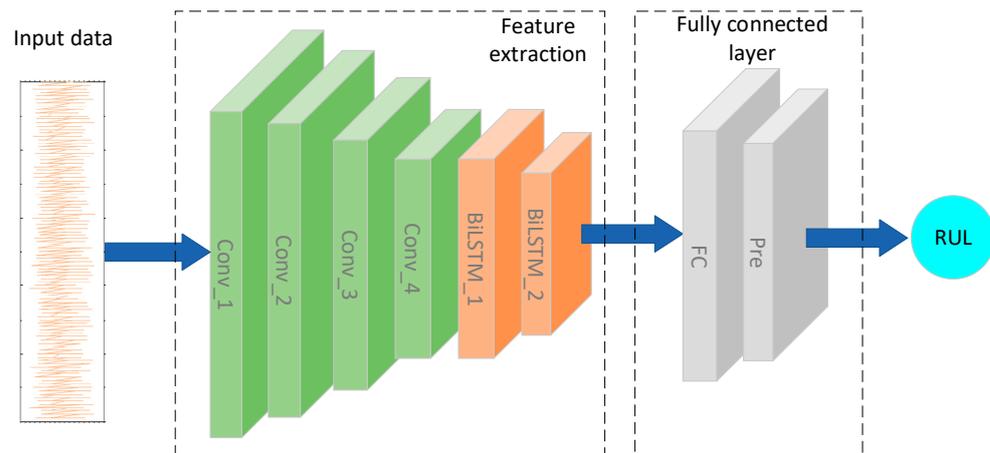


Figure 3. DCNN-BiLSTM structure diagram.

### 3.2. Domain Adaptation Method Based on MMD

Domain adaptation is a method of transfer learning. Domain adaptation is a machine learning algorithm that targets the distribution difference between source and target domains. A wide variety of domain adaptation methods aim to apply knowledge learned from the source domain to the target domain in the absence or few labels of the target domain by learning domain-invariant features of the source and target domains.

The MMD is the most widely used loss function in transfer learning, especially domain adaptation, and is mainly used to measure the distance between two different but similar distributions. Compared to other metrics, MMD can estimate nonparametric distances between various distributions and avoid the computation of intermediate process quantities. MMD maps the source and target domains to a Reproducing kernel Hilbert space (RKHS) and then calculates the distribution distance between the two domains. MMD is defined as:

$$MMD(X, Y) = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \varphi(x_i) - \frac{1}{n_t} \sum_{j=1}^{n_t} \varphi(y_j) \right\|_{\mathcal{H}}^2 \tag{10}$$

where  $\mathcal{H}$  represents the RKHS space,  $n_s$  and  $n_t$  represents the number of samples in the source domain and the target domain, respectively,  $\varphi(x) : X \rightarrow \mathcal{H}$  represents the mapping function from the original feature space to the RKHS, and then uses the kernel method to calculate the inner product to avoid high-dimensional complex operations, usually using a Gaussian kernel function, which represents for:

$$K(\mu, \nu) = e^{-\frac{\|\mu - \nu\|^2}{\sigma}} \tag{11}$$

where  $\mu$  and  $\nu$  represent different samples and  $\sigma$  is the width parameter of the function, which controls the radial range of the function.

In the case where there is a difference in the distribution between the source domain data and the target domain data, the MMD is added to the loss function to optimize the target. Therefore, the loss function of the pre-trained network model is defined as:

$$Loss = MSE\_loss + \lambda MMD\_loss \tag{12}$$

where  $MSE\_loss$  is the mean square loss function and  $\lambda$  represents the balance function,  $\lambda > 0$ .

The transfer learning in this paper is based on the method of domain adaptation. During the pre-training process, the source domain and target domain datasets are trained at the same time. The output value of the first fully connected layer of the DCNN-BiLSTM network model is used as the sample space for calculating the distribution distance between

the two domains, and finally, the pre-training model after domain adaptation is obtained. The training process is shown in Figure 4.

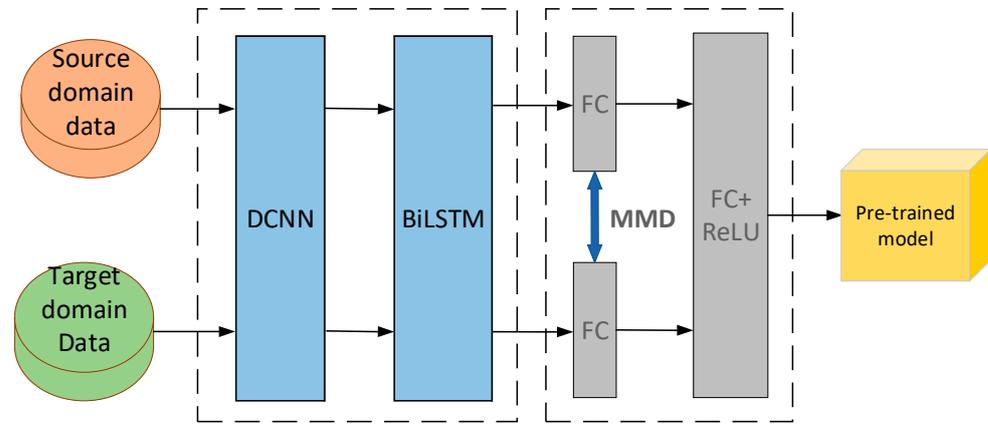


Figure 4. The pre-training framework based on domain adaptation.

### 3.3. Fine-Tune the Target Model

In order to shorten the training time of the target model, make the target model more adaptable to different operating conditions and environments, and improve the generalization ability. In this section, the Adam optimizer is used to fine-tune the pre-trained model. The flowchart of fine-tuning is shown in Figure 5. First, initialize the target model with the weights and parameters of the pre-trained model, then freeze the parameters of the feature extraction layers, including 4-layer CNN and 2-layer BiLSTM, and only update the parameters of the task-specific layer, i.e., the two-layer fully connected layer. Furthermore, to prevent overfitting, different learning rates are set for the two fully connected layers. Finally, a prediction model that is more suitable for the target domain task and has strong generalization ability is obtained by training.

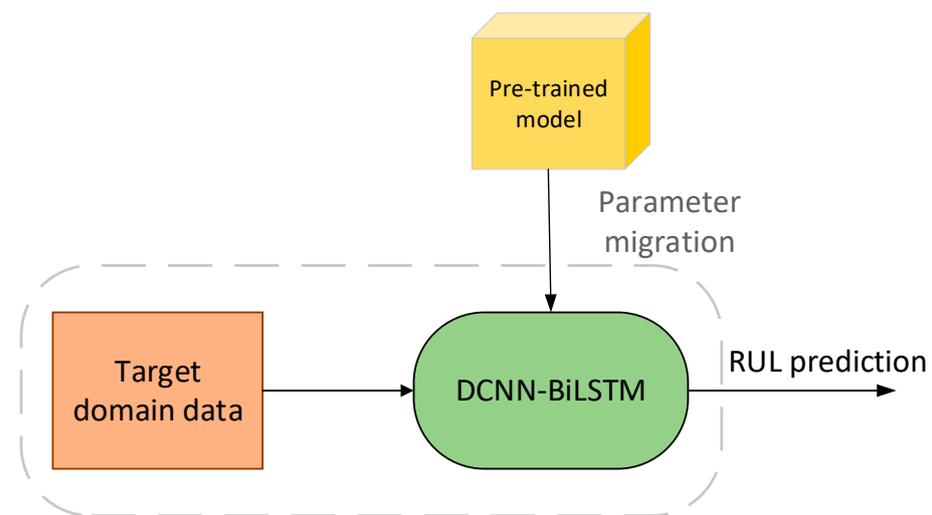


Figure 5. The flowchart of fine-tuning.

## 4. Experimental Results and Analysis

### 4.1. Dataset Description

The method in this paper was evaluated using the turbofan engine degradation data of the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset provided by NASA. The detailed information of the dataset is presented in Table 1. This dataset consists of four different sub-datasets with different operational conditions and

fault modes. Each sub-dataset contains time-series information collected by 21 sensors and 3 measurements of operational conditions. The training set and the test set have different numbers of degraded engines, each with a different degree of initial wear, and after the number of cycles increases, the engine slowly ages until it fails to work. The training set records the degradation process of the entire life cycle of the engine, while the test set only includes a certain moment before failure. The task is to predict the remaining useful life (RUL) of the engine units in the test set.

**Table 1.** Information of the C-MAPSS dataset.

Subdataset	FD001	FD002	FD003	FD004
Training set engine unit	100	260	100	249
Test set engine unit	100	259	100	248
Training set sample size	17,731	48,558	21,220	56,815
Test set sample size	100	259	100	248
Maximum life cycle	362	378	512	128
Operational conditions	1	6	1	6
Fault modes	1	1	2	2

Seven sensor values were observed to remain unchanged within the FD001 subset. In order to save computing resources, meaningless data is eliminated, and 14 sensors were obtained as 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, and 21.

#### 4.2. Data Processing

The original data is composed of data detected by multiple sensors. Different data sets have different sequence lengths, and the data dimensions are high and have different dimensions. Therefore, the min-max normalization method is used to unify the data into the range  $[-1,1]$ . Each measurement  $x_{i,j}$  is min-max normalized and can be expressed as [17]:

$$\tilde{x}_{i,j} = \frac{2(x_{i,j} - x_{min}^j)}{(x_{max}^j - x_{min}^j)} - 1 \quad (13)$$

where  $\tilde{x}_{i,j}$  represents the normalized data,  $x_{min}^j$  and  $x_{max}^j$  represents the minimum and maximum values of the data monitored by the  $j$ th sensor in one operating cycle, respectively.

In order to obtain more useful temporal information from the input data, the normalized data is subjected to time windowing. For continuous time-series data, a sliding time window is used to define data labels, and the size of the input model sequence is determined by the size of the time window.

The window of size  $N_w$  slides along the time series, and each time step slides  $l$  will feedback the data to the slider, which is used as the input of the prediction model, so the input size of the network is  $N_w \times m$ . To get more samples and reduce the risk of overfitting, the sliding time step is set to 1.

When the engine is running under normal conditions, taking the remaining operating cycle period as RUL, then we assume that RUL decreases linearly, using a piecewise linear function, choose 125 as the initial life period [18], and apply it to the training set and test set.

#### 4.3. Selection of Evaluation Indicators

To verify the effectiveness of the method in this paper, two functions were used as evaluation metrics, namely the Root Mean Square Error (RMSE) function and the Score function [19]. The RMSE function formula is:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2} \quad (14)$$

The formula for the Score function is:

$$Score = \begin{cases} \sum_{i=1}^N (e^{-\frac{\hat{y}_i - y_i}{15}} - 1), & \hat{y}_i - y_i < 0 \\ \sum_{i=1}^N (e^{\frac{\hat{y}_i - y_i}{10}} - 1), & \hat{y}_i - y_i \geq 0 \end{cases} \quad (15)$$

where  $\hat{y}_i$  and  $y_i$  represent the predicted value and the actual value of RUL, respectively.

RMSE reflects the degree of fit between the predicted life and the actual life, and the size of the Score measures the rationality of life prediction. The lower the values of RMSE and Score, the better the predictive ability of the model.

#### 4.4. Experimental Configuration and Parameters

All experiments in this paper are performed on a processor configured with 16 GB memory (RAM), NVIDIA GeForce TITAN XP graphics card, and Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz processor. The network model proposed in this paper is based on Python3.6 and the PyTorch deep learning framework. In the experiments in this paper, considering the influence of the sample size on the prediction accuracy and the influence of different operational conditions and fault modes, in order to improve the generalization ability of the RUL prediction model, according to the size of the data, we use FD002 and FD004 with the sufficient sample size in C-MAPPS as source domain datasets, and FD001 and FD003 datasets in C-MAPPS with insufficient sample size as target domains. We evaluate the performance of transfer learning in RUL prediction of the target domain and investigate how different working conditions and the number of samples of the source and target domain datasets affect the performance of the final prediction model. Therefore, set the experimental tasks as shown in Table 2.

**Table 2.** Transfer learning experiment tasks.

Source Domain	Target Domain	Operational Conditions	Fault Mode
FD002	FD001	6→1	1→1
	FD003	6→1	1→2
FD004	FD001	6→1	2→1
	FD003	6→1	2→2

#### 4.5. Model Prediction Results and Analysis

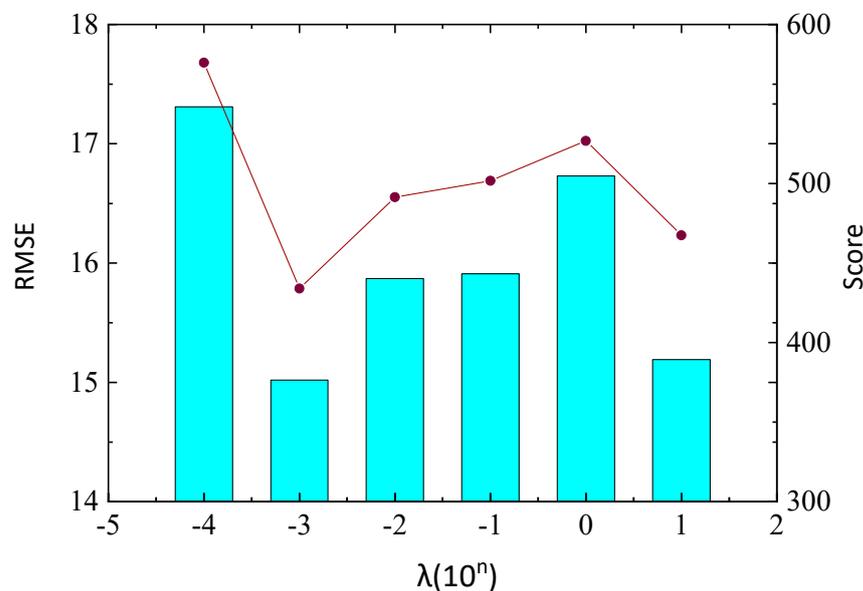
In order to compare the effectiveness of the transfer learning method proposed in this paper, the results of the method were compared with the experiments without transfer, as shown in Table 3. Source-Only refers to directly testing the target domain with the pre-trained model, Target-Only refers to training and testing only on the target domain.

**Table 3.** Compare transfer learning with no transfer.

Methods	FD002→FD001		FD002→FD003		FD004→FD001		FD004→FD003	
	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
Source-Only	84.63	5,355,207.6	51.39	362,392.1	41.45	18,345.5	42.65	14,591.2
Target-Only	15.84	534.13	14.66	281.12	17.28	473.87	16.32	349.55
TL	14.36	371.86	13.66	243.22	16.35	432.37	14.79	296.34

As can be seen from Table 3, the transfer learning algorithm proposed in this paper greatly improves the accuracy of the prediction model. Due to the influence of the difference in the distribution of the data set, the pre-training model of Source-Only was directly used for testing, and the effect was very bad. On the basis of the traditional Target-Only prediction method, the pre-training model after domain adaptation was loaded, and then the model was optimized in the target domain, and the prediction accuracy was improved. Take FD002→FD003 as an example, the RMSE increased by at least 6.82%, and the score function value increased by at least 13.48%.

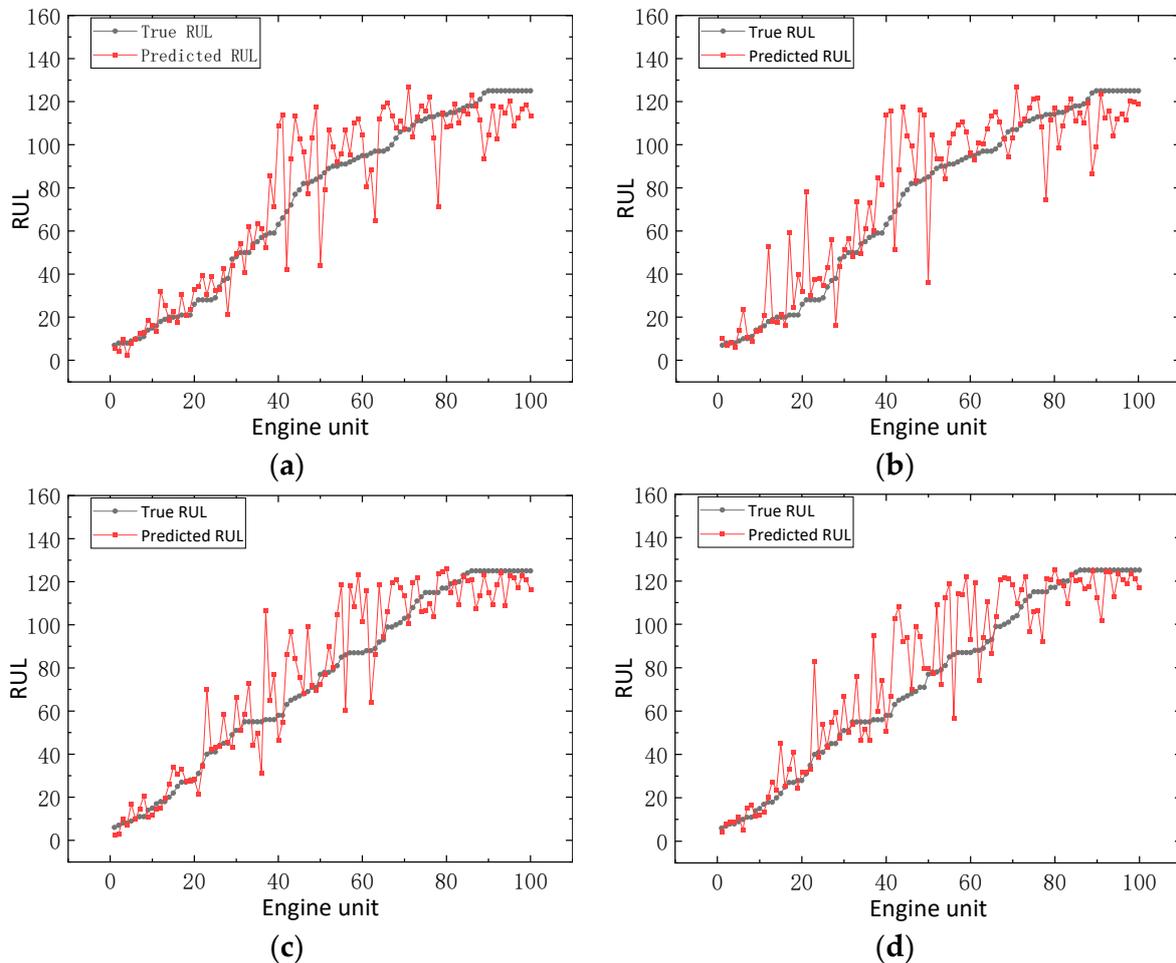
In the pre-training stage, the MMD item of the tuning process not only affected the prediction accuracy of the data set but also affected the matching degree of the conditional distribution. Therefore, the coefficient  $\lambda$  of MMD\_loss of the loss function had a greater impact on the adaptive effect. Taking FD002→FD001 as an example, when  $-1$  was used as the median value, a large number of comparative experiments were carried out by increasing or decreasing order of magnitude. As shown in Figure 6, the horizontal axis is the value size, and the vertical axis is the two evaluation indicators values of RMSE and Score. It can be seen that when  $\lambda = 0.001$ , both RMSE and Score achieve the minimum value, so the coefficient value  $\lambda$  of MMD\_loss in this experiment was 0.001.



**Figure 6.** The impact of different  $\lambda$  values on the prediction results.

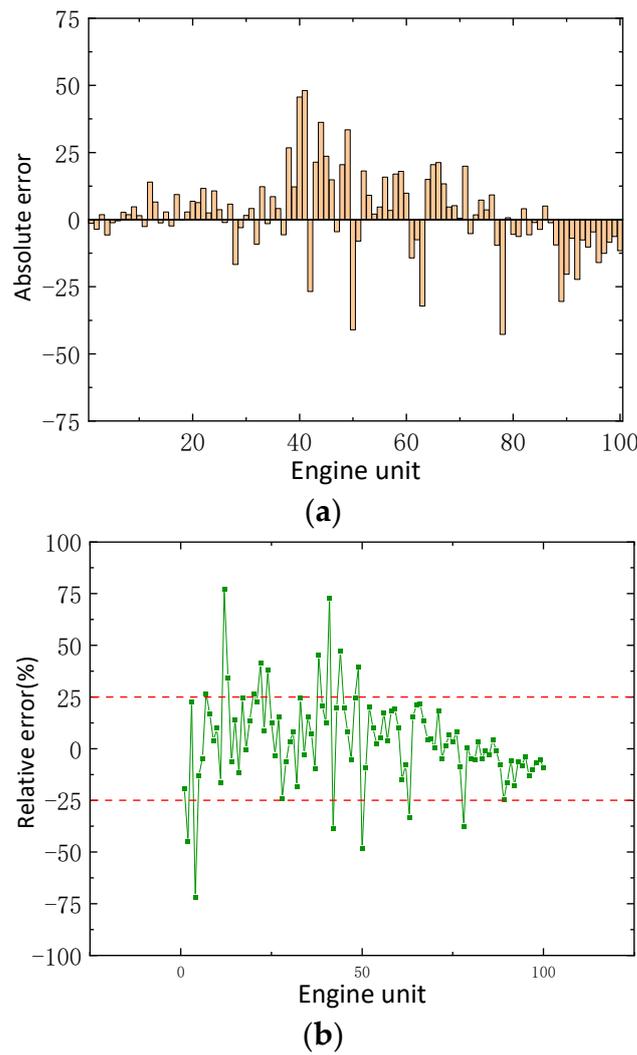
In order to compare the prediction effect of the DCNN-BiLSTM model based on transfer learning on the small sample data set, Figure 7 shows the prediction results of all the engine units on the test set sorted from small to large according to the RUL value of the four tasks of experiments. The horizontal axis represents the test engine unit, and the vertical axis represents the RUL. It can be seen from Figure 7 that the DCNN model could effectively extract the detailed features and similar features of the engine degradation, even if it is difficult to predict at the beginning of the operation. The value was also closer to

the set value of 125. As the running period increases, BiLSTM could effectively obtain the relationship between the time series before and after. Combining the functions of fusion model and domain adaptation, it can be seen from Figure 7 that its prediction trend was stable and could better fit the real degradation curve. Therefore, the transfer learning model proposed in this paper shows a good prediction effect.



**Figure 7.** RUL prediction results of four tasks of experiments. (a) FD001 Engine Prediction Results (FD002→FD001). (b) FD001 Engine Prediction Results (FD004→FD001). (c) FD003 Engine Prediction Results (FD002→FD003). (d) FD003 Engine Prediction Results (FD004→FD003).

Taking FD002→FD001 as an example, the error and relative error of all engines in FD001 are used to intuitively show the accuracy of RUL prediction with the method in this paper. The results are shown in Figure 8. It can be seen from Figure 8a that when the engine starts to run, the RUL value is relatively large, and the prediction error is relatively large. When the engine runs for a long time or is about to fail, the degradation information is more obvious, and the prediction performance is significantly enhanced. Under a limited sample, it is difficult to accurately predict the equipment life of one set of different working conditions with the sensor data of another set of working conditions. The method in this paper improves this problem to a certain level so that the relative error generally remains at  $[-25\%, 25\%]$  as the Figure 8b.



**Figure 8.** Error curve of RUL prediction results in task FD002→FD001. (a) Absolute error. (b)Relative error.

In order to verify the effectiveness of the DCNN-BiLSTM, the five state-of-the-art network models are used to compare the hybrid network DCNN-BiLSTM; the RUL prediction results are shown in Table 4. It can be observed that the DCNN-BiLSTM model performed significantly better than SVM, MLP, CNN, LSTM, and CNN-LSTM in datasets FD001 and FD003. The DCNN-BiLSTM adopts a multi-layer convolutional network structure and a bidirectional long and short-term memory network, which can extract spatial and temporal features in detail, strengthen the feature extraction ability, and effectively improve the prediction accuracy.

**Table 4.** The results of the hybrid network model in this paper are compared with other network models on the C-MAPSS dataset.

Methods	FD001		FD003	
	Score	RMSE	Score	RMSE
SVM [20]	7730.33	40.72	22,541.58	46.32
MLP [20]	560.59	16.78	479.85	18.47
CNN [7]	1290	18.45	1600	19.82
LSTM [8]	338	16.14	852	16.18
CNN-LSTM [21]	303	16.13	1420	17.12
DCNN-BiLSTM	532.16	15.98	365.41	15.63

To further verify the effectiveness and superiority of the proposed method in this paper, this method is compared with the advanced methods in recent years, and the comparison results with CORAL, WDGRL, DDC, ADDA, and RULDDA methods are shown in Table 5.

**Table 5.** The results of the methods in this paper are compared with other methods on the C-MAPSS dataset.

Methods	FD002→FD001		FD002→FD003		FD004→FD001		FD004→FD003	
	Score	RMSE	Score	RMSE	Score	RMSE	Score	RMSE
CORAL [22]	3590	24.43	23,071	42.66	154,842	51.44	6919	30.44
WDGRL [23]	157,672	15.24	19,053	41.45	45,394	42.01	77,977	18.18
DDC [24]	640	46.96	62,823	39.87	162,100	41.55	1623	44.47
ADDA [25]	689	19.73	11,029	37.22	43,794	37.81	1117	23.59
RULDDA [14]	2430	23.91	12,756	47.26	13,377	32.37	1679	23.31
DCNN-BiLSTM (TL)	371.86	14.36	243.22	13.66	432.37	16.35	296.34	14.79

From Table 5, we can see the proposed DCNN-BiLSTM (TL) method obtained substantially improved RMSE and Score prediction accuracy on all tasks. More specifically, RMSE and Score indicators on four tasks had reduced 5.77%, 63.26%, 49.49%, 18.65%, and 41.89%, 97.79%, 96.76%, and 73.47%, respectively, compared with the best result in the state-of-the-art methods. In addition, it can be observed that knowledge transfer between simple and complex datasets is challenging due to the large domain shift. For example, FD002→FD003 and FD004→FD001 are the transfer learning tasks of simple and complex datasets, and our proposed method obtained the greatest improvement and successfully aligned the two distant domains. The results show that the proposed transfer learning method could reduce the impact of operational conditions and fault modes on the RUL prediction accuracy of the target domain and effectively transfer the knowledge of the source domain with a large sample size, which is equivalent to data augmentation effectively for the target domain with small sample sizes. It improves the performance of the RUL prediction model. This is of great significance for equipment RUL prediction with small sample sizes in complex environments. Therefore, the proposed method is very promising in solving the small sample problem in the field of RUL prediction.

## 5. Conclusions

In the traditional data-driven RUL prediction method, the state detection data of the training set and the test set are required to have the same or similar distribution. However, due to different operational conditions, fault modes, and some force majeure factors in the actual working environment, it is generally difficult to obtain data sets that satisfy the same data distribution. In order to solve the problem that it is difficult to collect equipment operational data in some specific environments and the RUL prediction accuracy of equipment is not high, and the generalization ability is weak under different working conditions, this paper proposes a transfer learning-based RUL prediction method for small-sample equipment.

The method in this paper uses the DCNN-BiLSTM model to simultaneously train the source and target domain data and uses MMD to constrain the distribution difference between the two domains so as to realize the adaptation matching and feature alignment of the target domain samples and the source domain samples. The deep features are extracted to obtain a pre-trained model. Then, the network structure and parameters of the pre-trained model are transferred to the target domain for training by a fine-tuned transfer learning strategy, and the network is optimized. Finally, an RUL prediction model that is more suitable for the target domain data is obtained. When used on the C-MAPSS dataset, compared with other state-of-the-art methods, it verifies the effectiveness of the method

proposed in this paper for predicting the RUL of aero-engines. For the subsets FD002 and FD004 with complex operating conditions and sufficient sample data, the transfer learning method is used to solve the subsets FD001 and FD003 with single operating conditions and small data samples, and the effect is significantly improved.

In future research, more experiments will be conducted on different degradation datasets to demonstrate the reliability and generality of the proposed model. Furthermore, domain adaptation methods are applied to make unsupervised predictions on incomplete data of target domains with missing labels. Although the experiments in this paper have obtained good experimental results, it is still necessary to further optimize the network structure and parameters to improve the performance of the RUL model.

**Author Contributions:** Conceptualization, W.C. (Weizhao Chen) and W.C. (Wenbai Chen); methodology, W.C. (Weizhao Chen); validation, H.L. and Y.W.; formal analysis, H.L. and Y.W.; investigation, W.C. (Weizhao Chen) and W.C. (Wenbai Chen); resources, C.B. and Y.G.; writing—original draft preparation, W.C. (Weizhao Chen); writing—review and editing, W.C. (Weizhao Chen); supervision, W.C. (Wenbai Chen); project administration, W.C. (Wenbai Chen); funding acquisition, W.C. (Wenbai Chen), C.B. and Y.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by The Major Project of Scientific and Technological Innovation 2030 (2021ZD0113603), The Qin Xin Talents Cultivation Program, Beijing Information Science and Technology University (QXTCP A202102), and The General Project of Beijing Municipal Education Commission Scientific Research Program (KM202011232023).

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

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data of this paper came from the NASA Prognostics Center of Excellence, and the data acquisition website was: <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/#turbofan>, accessed on 10 February 2022.

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

## References

1. Hao, J.; Hu, Y.; Cui, N.; Han, F.; Xu, C. Research on GRU-BP for life prediction of key components in digital workshop. *J. Chin. Comput. Syst.* **2020**, *41*, 637–642.
2. Yurek, O.E.; Birant, D. Remaining useful life estimation for predictive maintenance using feature engineering. In Proceedings of the 2019 Innovations in Intelligent Systems and Applications Conference (ASYU), Izmir, Turkey, 31 October–2 November 2019; IEEE: Piscataway, NJ, USA; pp. 1–5.
3. Ahmadzadeh, F.; Lundberg, J. Remaining useful life estimation. *Int. J. Syst. Assur. Eng. Manag.* **2014**, *5*, 461–474. [[CrossRef](#)]
4. El-Thalji, I.; Jantunen, E. A summary of fault modelling and predictive health monitoring of rolling element bearings. *Mech. Syst. Signal Processing* **2015**, *60*, 252–272. [[CrossRef](#)]
5. Qin, S.J. Survey on data-driven industrial process monitoring and diagnosis. *Annu. Rev. Control.* **2012**, *36*, 220–234. [[CrossRef](#)]
6. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 1789–1819. [[CrossRef](#)]
7. Babu, G.S.; Zhao, P.; Li, X.L. Deep convolutional neural network based regression approach for estimation of remaining useful life. In *International Conference on Database Systems for Advanced Applications*; Springer: Cham, Switzerland, 2016; pp. 214–228.
8. Zheng, S.; Ristovski, K.; Farahat, A.; Gupta, C. Long short-term memory network for remaining useful life estimation. In Proceedings of the 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), Dallas, TX, USA, 19–21 June 2017; IEEE: Piscataway, NJ, USA; pp. 88–95.
9. Zhuang, F.Z.; Luo, P.; He, Q.; Shi, Z. Survey on transfer learning research. *J. Softw.* **2015**, *26*, 26–39.
10. Fu, B.; Wu, Z.; Guo, J. Remaining Useful Life Prediction under Multiple Operation Conditions Based on Domain Adaptive Sparse Auto-Encoder. In Proceedings of the 2020 IEEE International Conference on Prognostics and Health Management (ICPHM), Detroit, MI, USA, 8–10 June 2020; IEEE: Piscataway, NJ, USA; pp. 1–8.
11. Li, X.; Zhang, W.; Ding, Q.; Sun, J.Q. Multi-layer domain adaptation method for rolling bearing fault diagnosis. *Signal Processing* **2019**, *157*, 180–197. [[CrossRef](#)]
12. Ragab, M.; Chen, Z.; Wu, M.; Foo, C.S.; Kwok, C.K.; Yan, R.; Li, X. Contrastive adversarial domain adaptation for machine remaining useful life prediction. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5239–5249. [[CrossRef](#)]
13. Miao, M.; Yu, J. A Deep Domain Adaptive Network for Remaining Useful Life Prediction of Machines under Different Working Conditions and Fault Modes. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–14. [[CrossRef](#)]

14. da Costa, P.R.D.O.; Akçay, A.; Zhang, Y.; Kaymak, U. Remaining useful lifetime prediction via deep domain adaptation. *Reliab. Eng. Syst. Saf.* **2020**, *195*, 106682. [[CrossRef](#)]
15. Lv, H.; Chen, J.; Pan, T. Sequence Adaptation Adversarial Network for Remaining Useful Life Prediction Using Small Data Set. In Proceedings of the 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), Warwick, UK, 20–23 July 2020; IEEE: Piscataway, NJ, USA; Volume 1, pp. 115–118.
16. Yao, K.; Cohn, T.; Vylomova, K.; Duh, K.; Dyer, C. Depth-gated LSTM. *arXiv* **2015**, arXiv:1508.03790.
17. Chen, W.; Liu, H.; Chen, Q.; Wu, P. A Prediction Method for the RUL of Equipment for Missing Data. *Complexity* **2021**, *2021*, 2122655.
18. Listou Ellefsen, A.; Bjørlykhaug, E.; Æsøy, V.; Ushakov, S.; Zhang, H. Remaining usefullife predictions for turbofan engine degradation using semi-supervised deep archi-tecture. *Reliab. Eng. Syst. Saf.* **2019**, *183*, 240–251. [[CrossRef](#)]
19. Zhang, A.; Wang, H.; Li, S.; Cui, Y.; Liu, Z.; Yang, G.; Hu, J. Transfer learning with deep recurrent neural networks for remaining useful life estimation. *Appl. Sci.* **2018**, *8*, 2416. [[CrossRef](#)]
20. Zhang, C.; Lim, P.; Qin, A.K.; Tan, K.C. Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 2306–2318. [[CrossRef](#)] [[PubMed](#)]
21. Kong, Z.; Cui, Y.; Xia, Z.; Lv, H. Convolution and long short-term memory hybrid deep neural networks for remaining useful life prognostics. *Appl. Sci.* **2019**, *9*, 4156. [[CrossRef](#)]
22. Sun, B.; Feng, J.; Saenko, K. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*; Springer: Cham, Switzerland, 2017; pp. 153–171.
23. Shen, J.; Qu, Y.; Zhang, W.; Yu, Y. Wasserstein distance guided representation learning for domain adaptation. In Proceedings of the Association Advancement Artificial Intelligence (AAAI), New Orleans, LA, USA, 2–7 February 2018; pp. 4058–4065.
24. Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep domainconfusion: Maximizing for domain invariance. *arXiv* **2014**, arXiv:1412.3474.
25. Tzeng, E.; Hoffman, J.; Saenko, K.; Darrell, T. Adversarial discrim-inative domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7167–7176.