mathematics

MDPI

Article

Remaining Useful Life Prediction Based on Multi-Representation Domain Adaptation

Yi Lyu [1,2,*], Qichen Zhang [2], Zhenfei Wen [2] and Aiguo Chen [2]

[1] School of Computer, University of Electronic Science and Technology of China Zhongshan Institute, Zhongshan 528400, China
[2] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
* Correspondence: lvyi913@zsc.edu.cn

Abstract: All current deep learning-based prediction methods for remaining useful life (RUL) assume that training and testing data have similar distributions, but the existence of various operating conditions, failure modes, and noise lead to insufficient data with similar distributions during the training process, thereby reducing RUL prediction performance. Domain adaptation can effectively solve this problem by learning the cross-domain invariant features of the source domain and target domain to reduce the distribution difference. However, most domain adaptive methods extract the source domain and target domain features into a single space for feature alignment, which may leave out effective information and affect the accuracy of prediction. To address this problem, we propose a data-driven approach named long short-term memory network and multi-representation domain adaptation (LSTM-MRAN). We standardize and process the degraded sensor data with a sliding time window, use LSTM to extract features from the degraded data, and mine the time series information between the data. Then, we use multiple substructures in multi-representation domain adaptation to extract features of the source domain and target domain from different spaces and align features by minimizing conditional maximum mean difference (CMMD) loss functions. The effectiveness of the method is verified by the CMAPSS dataset. Compared with methods without domain adaptation and other transfer learning methods, the proposed method provides more reliable RUL prediction results under datasets with different operating conditions and failure modes.

Keywords: remaining useful life; sliding time window; long short-term memory network; multi representation domain adaptation

MSC: 93-10

Citation: Lyu, Y.; Zhang, Q.; Wen, Z.; Chen, A. Remaining Useful Life Prediction Based on Multi-Representation Domain Adaptation. *Mathematics* **2022**, *10*, 4647. https://doi.org/10.3390/math10244647

Academic Editor: Ioannis G. Tsoulos

Received: 8 November 2022
Accepted: 6 December 2022
Published: 8 December 2022

1. Introduction

With the development of science and technology, the levels of intelligence, integration and complexity of industrial systems are increasingly higher, and the traditional fault diagnosis and maintenance support technology encounter difficulty in gradually adapting to new requirements. Prognostics and health management (PHM) technology opens up a new path for improving system reliability and safety and plays an important role in product system health monitoring, prediction and management. PHM technology can predict and manage the possible risks of the system in the future, reduce maintenance support costs and improve the safety and reliability of equipment [1]. In PHM, remaining useful life (RUL) prediction is one of the core research problems. Accurate RUL prediction enables stakeholders to assess the health of product equipment and plan future maintenance actions, which is the theoretical basis for preventive and predictive maintenance in PHM.

RUL prediction methods can be divided into two categories: based on a degradation model and based on a data-driven model [2]. For the current increasingly complex and intelligent systems, establishing an accurate degradation model is difficult because of its

complex structure, diverse failure modes, and uncertainty of operating conditions. The method based on degraded data does not require prior knowledge and a complex physical modeling process, and gradually becomes the mainstream method for RUL prediction. The data-driven methods need to analyze the degradation data monitored by a large number of sensors and mine the internal laws of product performance degradation to predict the RUL of equipment. Owing to its powerful data analytical ability and function mapping ability, machine learning (ML) is widely used in RUL prediction based on degradation data, such as multilayer perceptron (MLP) [3], radial basis function (RBF) [4] and support vector machine (SVM) [5] methods. However, the traditional RUL prediction method of ML is usually a shallow model. Extracting features and expressing functional relations is difficult when predicting multivariate complex time series. Deep learning technology has a strong feature extraction ability, which provides a solution for training massive data. Badu et al. [6] first applied deep convolutional neural network (CNN) to RUL prediction, using two convolutional layers and two pooling layers to extract features of degraded data. Li et al. [7] proposed a multivariate equipment RUL prediction based on deep CNN and adopted a sliding time window method to obtain samples to better extract features. However, these methods ignore the time correlation between sensor monitoring degradation data. Due to its special network structure, the recurrent neural network (RNN) can retain the state information at the last time on the hidden layer, so it has more advantages in time series feature extraction. Liu et al. [3] used adaptive RNN to predict the RUL of lithium batteries. Malhi et al. [8] used a competitive learning method to cluster the input data and input the processed data into the RNN network for training, which improved the prediction performance. However, when dealing with long-term monitoring sequences, the traditional RNN encounters problems such as gradient disappearance and gradient explosion, which affect the prediction accuracy. Wu et al. [9] used the LSTM model to predict the RUL of aircraft engines and used dropout technology to improve the generalization ability of LSTM, effectively avoiding problems such as the disappearance of gradients in traditional RNNs. Zhang et al. [10] proposed an RUL prediction method based on BiLSTM, where the forward and backward paths are independently calculated and outputs are connected in series, which effectively smooths the prediction results. Kong et al. [11] proposed a fusion algorithm of CNN and LSTM, and applied it in RUL prediction to learn temporal and spatial features and improve the prediction accuracy.

Although the aforementioned methods have achieved relatively accurate RUL results, a key problem remains. In most of the equipment RUL predictions, the test and training sets are usually assumed to be from the same working condition and obey the same distribution, so the model has accurate prediction results only under the same working condition. However, in the process of actual product use, most of the products have variable working conditions, and the distribution of sensor monitoring data shows a certain difference, which leads to a sharp decline in the accuracy of RUL prediction. To solve these problems, we need to use models that can adapt to different input characteristics, data distributions, and failure modes. Based on the TCN-RSA framework, Cao et al. [12] used the residual self-attention mechanism to shape the internal features, and realized the accurate RUL prediction of the bearing under different working conditions. Unsupervised domain adaptation (UDA) can generalize the knowledge learned from the source domain to the target domain, learn cross-domain invariant features, and effectively reduce the distribution difference between the source and target domains [13]. UDA mainly achieves distribution alignment through methods such as distance measurement [14], feature reconstruction [15], and adversarial learning [16], reducing the direct distribution difference between the target domain data and source domain data so that the models can be migrated seamlessly. At present, domain adaptation has had several applications in PHM. Lu et al. [17] used MMD for the first time to realize domain adaptation in fault diagnosis. Li et al. [18] proposed multi-layer domain adaptation (MLDA), using MK-MMD and pseudo-labels. Learning to achieve cross-domain fault diagnosis, Han et al. [19] used the adversarial learning method to reduce the distribution difference between the source and target domains to achieve

cross-domain fault diagnosis. All of the aforementioned domain adaptive learning methods are intended to solve the problem of fault classification, while for regression problems such as RUL prediction, the method of transfer learning is also widely used. Zhang et al. [20] used CNN as a feature extractor and MMD as a domain adaptation loss function to achieve great results in RUL prediction under different operating conditions. Cao et al. [21] applied BiLSTM network to transfer learning, proposed TBiLSTM network structure, and used MMD as the domain adaptation loss function to solve the problem of bearing distribution difference under different working conditions. Da Costa et al. [22] combined LSTM with adversarial training DaNN to align the characteristics of the source domain and the target domain by maximizing the domain discrimination loss function. Fu et al. [23] combined the above two methods and simultaneously used MMD and DaNN to align the characteristics of the source domain and the target domain, and achieved remarkable results. Although these methods effectively improve the accuracy of RUL prediction under different working conditions, due to the distribution of the source domain and target domain being complex, the feature extracted by a single structure can only contain part of the information. Therefore, in order to make full use of the information on the source and target domains and extract more comprehensive cross-domain invariant features, this paper proposes an RUL prediction model based on LSTM and multi-representation domain adaptation (MRAN) to represent the original data more comprehensively and improve the domain adaptability. The temporal and spatial features of the source domain and target domain are fully extracted by LSTM and multiple convolution substructures, and the distribution differences between domains are reduced from different feature spaces by minimizing CMMD. The extracted features after the final training are used as the degradation features of the target domain for life prediction. The structure can extract cross-domain invariant features from different spaces, effectively improving the prediction accuracy of RUL under different working conditions. The performance of the model is verified by using turbine engine data sets under different operating conditions and fault modes.

The main contributions of this paper are summarized as follows:

(1) In order to adapt to more application scenarios, the proposed structure does not need to obtain the target domain label, but obtains the pseudo label through the label classifier.

(2) In order to make full use of the input information, we use the LSTM-MRAN structure to extract the source domain and target domain features from multiple scales.

(3) Align feature distributions from multiple substructures simultaneously by minimizing conditional maximum mean dispersion (CMMD).

The remainder of this article is organized as follows. Section 2 introduces the network structure of the proposed LSTM-MRAN model in detail. In Section 3, a case study is conducted using the aircraft turbofan engine data from NASA, furthermore, the effectiveness and advantage of the proposed model are validated by carrying out a comparison with other methods based on the same experimental data sets. Section 4 concludes the study and points out the future work direction.

2. Proposed Structure

This study proposes an LSTM-MRAN structure to predict RUL using a domain-adaptive learning approach. The structure consists of four parts: (1) feature extractor $D = f_d(x; \theta_d)$, extract temporal features of degraded data for domain adaptation, label classification, and RUL prediction; (2) label classifier $C = f_c(x; \theta_c)$, train source domain data and generate pseudo tags of the target domain; (3) domain adaptor M, align source and target domain characteristics and; (4) RUL predictor $L = f_l(x; \theta_l)$, output RUL prediction value and evaluation function value. The network structure of the four parts and the training process of LSTM-MRAN is shown in Figure 1. Since the target domain data does not have a real RUL label, the source domain data is input to pre-train the network model, and then the target domain data is input to obtain a pseudo classification label, which is applied to the domain adaptation process. The third part is the domain adaptation part. Multiple

source domain and target domain features are extracted through the convolution layer of different convolution kernels of LSTM-MRAN, and the distribution difference between the source domain and target domain is reduced by minimizing CMMD. The inputs of the framework are the source domain data and target domain data after normalization and sliding window processing, and the outputs are the RUL prediction value and evaluation function value. The model flow is shown in Figure 2 and Algorithm 1. Figure 2 shows the overall framework of the whole structure. The source domain data and target domain data processed by normalization and sliding windows extract multiple subrepresentations through multiple substructures of the feature extractor. The feature extractor consists of two parts, the LSTM module processes time series data and the multiple representation module extracts different representation structures through different convolution kernels. The extracted subrepresentations are used as the input of the domain adaptation module, and the CMMD loss function is minimized through backpropagation to reduce the distribution difference between domains. Finally, the RUL predictor receives the cross-domain invariant feature from the domain feature extractor to predict the RUL value. Because the target domain does not have enough labels, the label classifier outputs the target domain pseudo labels after pre-training with the source domain data. Then, we will describe in detail the four substructures and the joint loss function in the framework.

Algorithm 1: Domain adaptation and RUL prediction algorithm.

Input: historical degradation data $X_{s,n}$

1. The data preprocessing strategy processes the original degraded data and divides the data set into a training set and a test set.
2. Slide the data into the window for processing.
3. Input the source domain pre-training feature extractor and label the classifier with labels.
4. Input the target domain data of the training set and obtain the pseudo label through the feature extractor and label classifier.
5. Input the training set (real label source domain data + pseudo label target domain data) to the domain adaptor, and use CMMD to calculate the distribution difference.
6. Gradient descent algorithm minimizes CMMD, and backpropagation updates feature extractor parameters.
7. Input test set data into the trained structure.

Output: predicted RUL value and evaluation function.

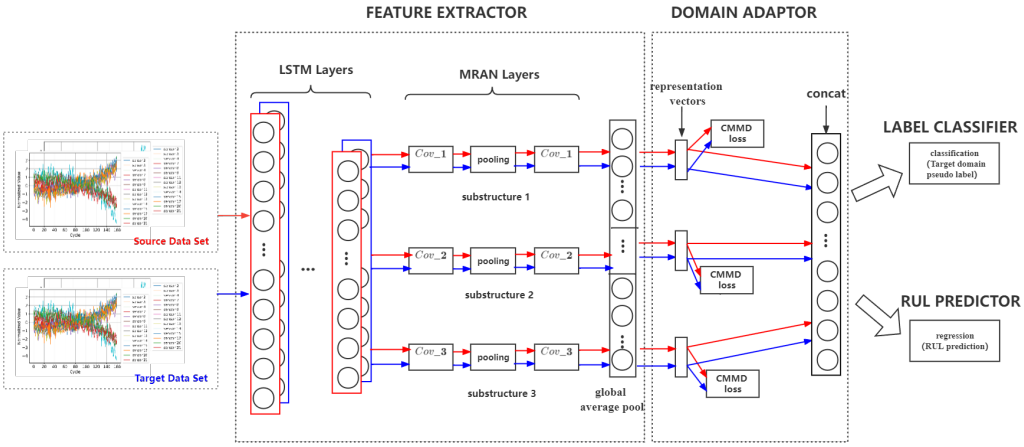

Figure 1. Network structure of LSTM-MRAN.

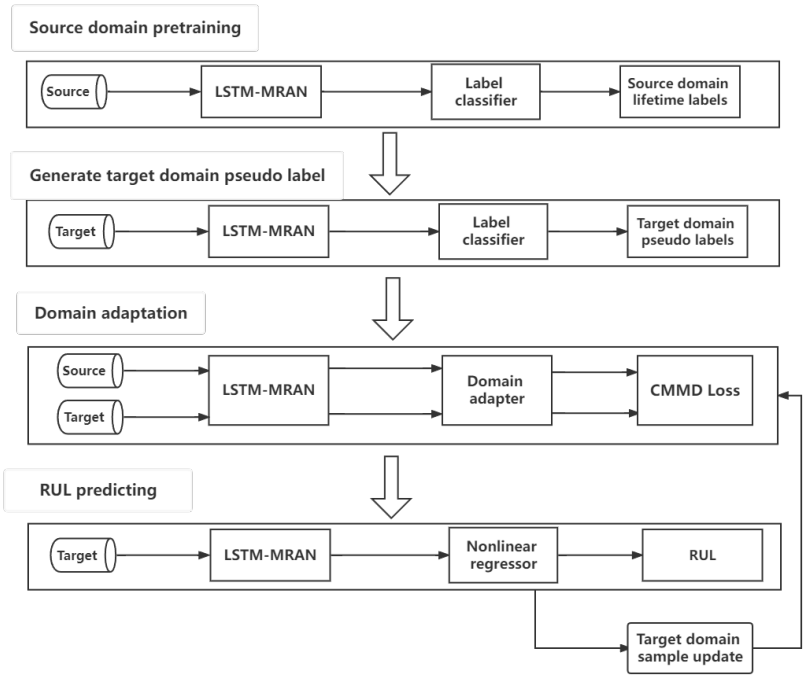

Figure 2. Structure of LSTM-MRAN.

2.1. Feature Extractor

In the process of predicting the RUL of products based on degradation data, historical data have an important effect on the performance and state of products at the current time. The long short-term memory network (LSTM) is improved by RNN, it solves the problem that RNN cannot deal with remote dependence. The LSTM unit consists of three gate structures: forget gate, input gate, and output gate. The structure of the LSTM unit is shown in Figure 3, the cell state update includes the following steps: the forget gate is responsible for determining which information will be discarded from the cell state, then the input gate determines which information currently input will be retained in the cell state, and finally the output gate determines which information the cell will output. In this way, the problems of gradient disappearance or explosion are effectively overcome.

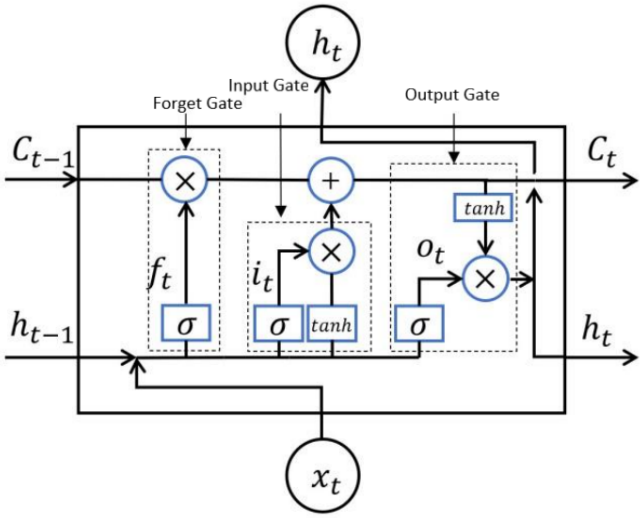

Figure 3. Structure of LSTM unit.

The related formula is shown as follows:

Forget gate:

$$f_t = \sigma\Big(W_f \cdot [h_{t-1}, x_t] + b_f\Big), \tag{1}$$

where W_f is the correlation weight matrix, b_f is the offset term, σ is the activation function, the input of the current time consists of two parts, the model output of the previous time is h_{t-1} and the model input of the current time is x_t. The activation function selects whether the information is forgotten or enters the network to participate in the subsequent update of network parameters.

Input gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t, \tag{4}$$

where, W_i, W_C is the correlation weight matrix, b_i, b_c is the offset term, σ, tanh is the activation function. The input of the input gate is the same as that of the forget gate. After the nonlinear selection of the two activation functions, its output consists of the output of the input gate at the previous time C_t, the output of the forget gate at the current time f_t and the current output $\tilde{C}_t$.

Output gate:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \times \tanh(C_t), \tag{6}$$

where, W_o is the correlation weight matrix, b_o is the offset term, σ, tanh is the activation function. The output of the input gate reduces the value to the interval $(-1, 1)$ through the tanh activation function and determines the final output of the model together with the first two inputs h_{t-1}, x_t.

Therefore, to fully mine the temporal information between the data, we use the three-layer LSTM network as the feature extractor. Each layer of the LSTM network has 100 neurons and the Relu activation function is used. The temporal information of the input data is mined by LSTM and the preliminary features are extracted. The extracted preliminary features are used as the input of domain adaptation. The structure of the feature extractor is shown in Figure 4.

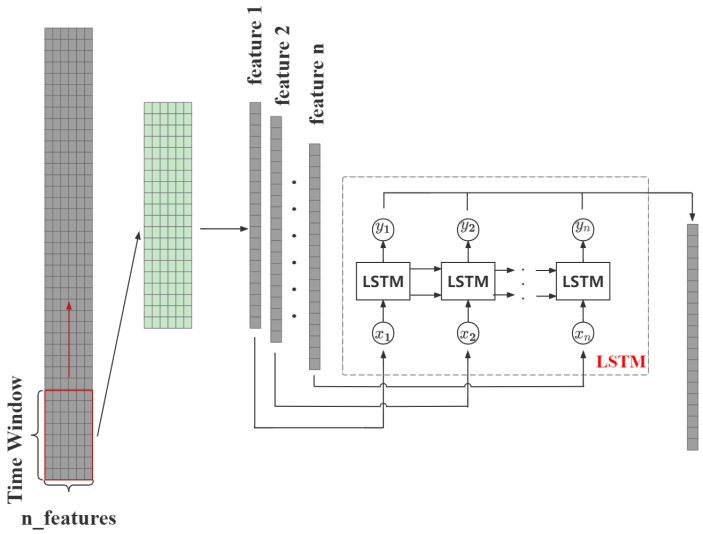

Figure 4. Structure of LSTM.

2.2. Domain Adaptor

Domain adaptation is a kind of transfer learning. It is an ML algorithm for solving the distribution offset of the source domain set and target domain. The purpose is to apply

the classifier learned from the source domain to the target domain by learning the domain invariant features of the source and target domains, to improve the prediction ability of the model to the target domain.

2.2.1. Multi-Representation Adaptive Network

In unsupervised domain adaptation learning, a source domain $D_s = \{(x_i^s, y_j^s)\}_{i=1}^{n_s}$ and a target domain $D_t = \{x_j^t\}_{j=1}^{n_t}$ are defined, where the source domain has n_s labeled data and the target domain has n_t unlabeled data. The label values of D_s and D_t are in the same range, and the data distribution of the two domains is not equal, that is, $P_s(x_s) \neq P_t(x_t)$. The purpose of domain adaptation learning is to use the model to learn knowledge from the source domain and improve the prediction ability of the target domain. As shown in Figure 5, the main idea is to find the cross-domain invariant characteristics in line with the common distribution, and shorten the distance between the source and target domains. Domain adaptive neural network (DaNN) [24] uses maximum mean discreteness (MMD) to extract domain invariant features. Tzeng et al. [25] proposed DDC neural network to further shorten the distance between the two domains by combining neural network and MMD. Long et al. [26] used MK-MMD to replace the original single-core MMD and proposed DAN using multiple cores to construct the total core. These studies are from the perspective of reducing the distribution differences between domains, and transfer by designing better distribution measurement differences.

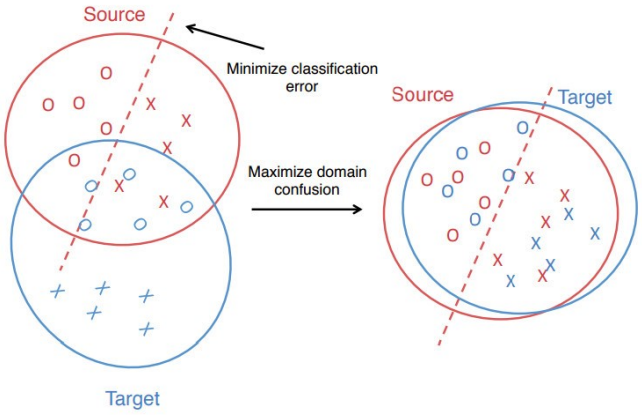

Figure 5. Domain Adaptation

However, in practical problems, the data distribution of the source and target domains usually have a complex structure. Using only the feature information extracted from a single structure is not enough, and even negative migration may occur. To solve this problem, we use MRAN to align the conditional distributions of multiple representations and extract domain-invariant features. The feature alignment method and MRAN frame structure are shown in Figures 6 and 7, respectively. The features are mapped to multiple feature spaces through multiple substructures and aligned in multiple feature spaces. Compared with a single representation that contains only part of the information, multiple representations may contain more information.

In this study, the input vector of MRAN is the feature vector of the source domain and target domain extracted by LSTM, which is aligned in different feature spaces through the multi-scale convolution kernel in MRAN. The width of the multi-scale convolution kernel is the width of input data, and the length is the convolution kernel $\{F_1, F_2, \ldots, F_{n_r}\}$ with different scales. In order to make the length of multiple subrepresentation vectors of MRAN output consistent, the method of zero-filling is used. After each convolution operation, the activation function is used to increase the nonlinearity of the model. The features extracted by multi-scale convolution kernel can be expressed as:

$$x_m = \{\phi(\omega_{m,1}^T \times \sigma(X_{i:i+Nw-1}) + b_{m,1}), \\ \phi(\omega_{m,2}^T \times \sigma(X_{i:i+Nw-1}) + b_{m,2}), \\ \ldots, \phi(\omega_{m,n_r}^T \times \sigma(X_{i:i+Nw-1}) + b_{m,n_r})\}, \tag{7}$$

where ϕ is the nonlinear activation function, $\omega_{m,n}$ and $b_{m,n}$, respectively, represent the weight and bias term of the nth convolution kernel, and σ is the LSTM function. Finally, the features obtained from each substructure are aligned using domain adaptive alignment and spliced together as the input of RUL prediction.

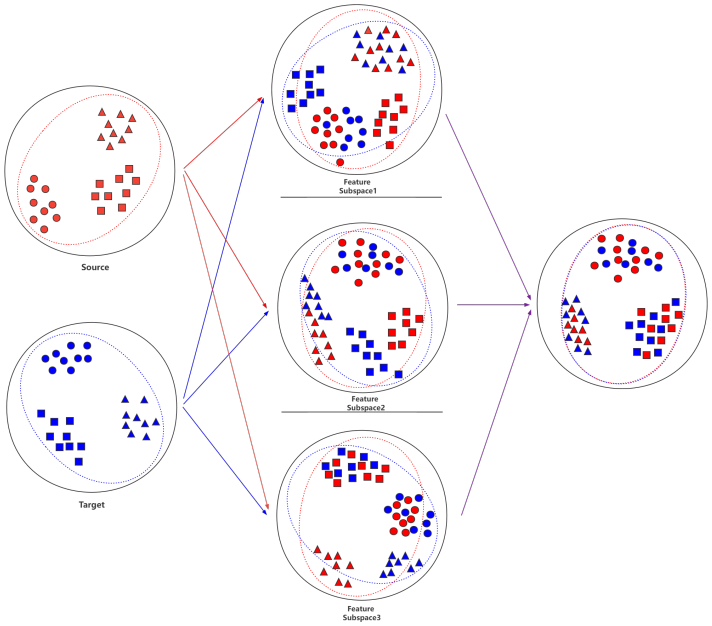

Figure 6. Feature alignment method of MRAN.

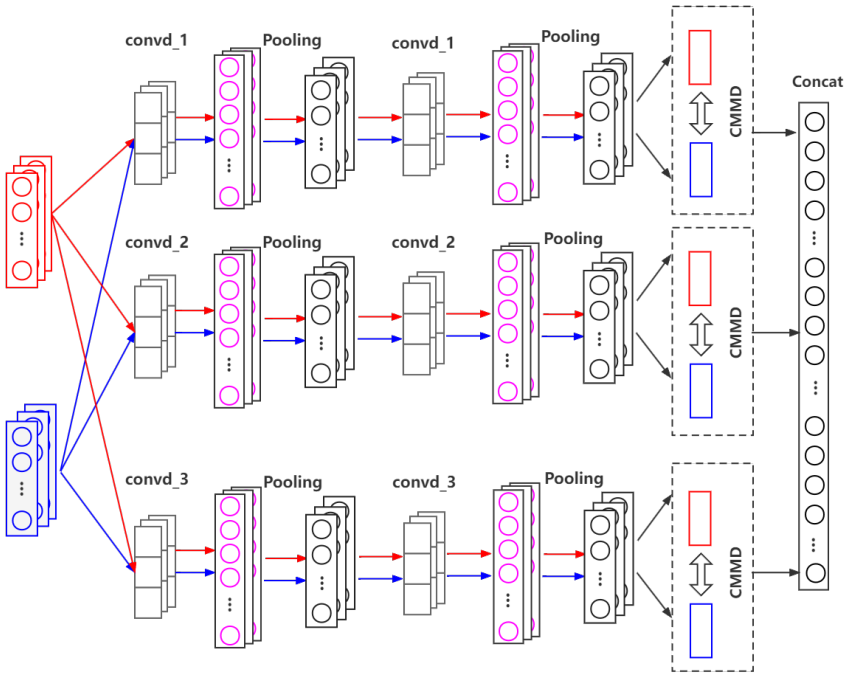

Figure 7. MRAN network structure.

2.2.2. Conditional Maximum Mean Discrepancy

In single representation domain adaptation, MMD is one of the most commonly used and widely used loss functions, which is mainly applied to measure the distance between two different but related distributions. The distance between the two distributions is defined as:

$$\hat{d}_H(X_s, X_t) = \| \frac{1}{n_s} \sum_{x_i \in D_{X^s}} \Phi(X_i) - \frac{1}{n_t} \sum_{x_j \in D_{X^t}} \Phi(X_j) \|_H^2. \tag{8}$$

In this study, as no labeled data exist in the target domain, it cannot directly match the conditional distribution of the target domain. Thus, we use CMMD to adapt the source and target domains. CMMD is derived from MMD, which divides data into different parts based on true source domain data and pseudo-labeled target domain data. Using class-conditional probability according to the Bayesian formula, we explore the sufficient statistics of class-conditional distributions $P(x_s|y_s = c)$ and $Q(x_t|y_t = c)$ instead with respect to each class $c \in \{1, \ldots, C\}$ [27]. Then, we modify MMD to measure the distance between the class-conditional distributions $P(x_s|y_s = c)$ and $Q(x_t|y_t = c)$ and combine the MMD distance on all categories to obtain CMMD. CMMD is shown in Equation (9).

$$\begin{aligned} \hat{d}_H(X_s, X_t) = \frac{1}{C} \sum_{c=1}^{C} \| & \frac{1}{n_s^{(c)}} \sum_{x_i^{s(c)} \in D_{X^s}^{(c)}} \Phi(X_i^{s(c)}) \\ & - \frac{1}{n_t^{(c)}} \sum_{x_j^{t(c)} \in D_{X^t}^{(c)}} \Phi(X_j^{t(c)}) \|_H^2. \end{aligned} \tag{9}$$

By minimizing the CMMD, we align the same class-conditional distribution to reduce the overall distance between the source and target domains. Thus, the optimization objective of the domain adaptor is:

$$\min_{\theta_g} L_m = \hat{d}_H(X_s, X_t), \tag{10}$$

where L_m is the loss function of the domain adaptor.

Therefore, we use MRAN as the domain adaptor and apply stochastic gradient descent (SGD) minimization domain adaptation loss function CMMD to align the source domain and target domain feature distribution as well as backpropagation to update the feature extractor and domain adaptor parameters.

2.3. Label Classifier

The label classifier is used to output the pseudo label of the target domain. Since RUL prediction is a continuous regression problem, we divide the product lifecycle data into 10 parts through the discrete method [28], and then mark them from 1 to 10 as classification labels, corresponding to 10 categories. As shown in Equation (11) and Figure 8.

$$y_c = \left\lceil \frac{t_r}{T_u} \times 10 \right\rceil = \left\lceil \frac{t_r}{t_r + t_u} \times 10 \right\rceil, \tag{11}$$

where, y_c is the classification label, t_r is the remaining use time, t_u is the used time, and T_u is the total use time.

The label classifier is composed of a full connection layer and a softmax layer. The full connection layer uses Relu activation, the softmax layer outputs prediction labels, the loss function adopts cross-entropy, and the loss function is shown in Equation (12).

$$L_c(f_c, f_g) = -\sum_{i=1}^{n} y_i^s \log f_c(f_g(x_i^s)). \tag{12}$$

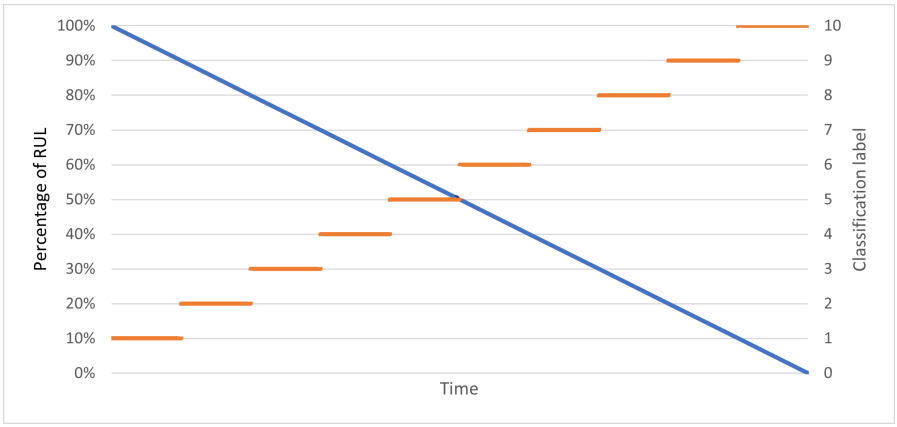

Figure 8. RUL and Classification label.

2.4. RUL Predictor

The multiple feature representations obtained from MRAN are contacted together as the input of the RUL predictor. The RUL predictor is used to output the final RUL prediction value and compare it with the real RUL value to output the loss function. RUL predictor updates parameters by backpropagation of pre-training with source domain datasets, and executes random gradient descent algorithm to minimize the overall loss in an iterative manner. The RUL predictor is composed of three full connection layers, the number of neurons is 32, 16, and 1, respectively, and ReLU is used as an activation function. The evaluation metric commonly used in RUL prediction include RMSE (root mean square error), MAE (mean absolute error), and score function.

2.5. Joint Loss

The joint loss function of the LSTM-MRAN structure consists of two parts; label classifier loss function L_c and domain adaptor loss function L_d. The label classifier uses the cross entropy function, and the loss function is shown in Equation (13).

$$L_c(f_c, f_g) = -\sum_{i=1}^{n} y_i^s \log f_c(f_g(x_i^s)). \tag{13}$$

The training goal of LSTM-MRAN is to minimize the classification error on the dataset of the source domain and the conditional maximum mean dispersion between the source domain and the target domain. The total loss function can be expressed as:

$$L(\theta_g, \theta_c, \theta_y) = L_c(\theta_g, \theta_c) + L_y(\theta_g, \theta_y) + \lambda L_d(\theta_g). \tag{14}$$

Based on the above total loss function, the optimization target of parameters θ_g, θ_c can be given as Equation (15).

$$\hat{\theta_g}, \hat{\theta_c}, \hat{\theta_y} = arg \min_{\theta_g, \theta_c, \theta_y} L(\theta_g, \theta_c, \theta_y). \tag{15}$$

3. Experiment

3.1. Experiment Datasets

This study uses the CMAPSS turbine engine dataset [29], which consists of four different subsets, which in turn contain information from 21 sensors and 3 operation settings. Each subset is divided into a training set and a test set. There is a complete track from running to a fault in the training set. In the test set, the trajectory ends before the equipment failure. In addition, the dataset collects fault information from multiple engines under various operating conditions and fault modes.

The details of the four sub datasets are shown in Table 1. The four subsets are FD001, FD002, FD003, and FD004. Among them, the engine working condition in the FD001

dataset is the simplest, with only one operating condition and one failure mode. The FD004 working condition is the most complex, with 6 operating conditions and 2 failure modes. The division of the experimental dataset is shown in Table 2. Two prediction tasks of migration learning are used to evaluate the performance effect of the model, namely, using FD004 pre-adaptation to learn the other three data set conditions and FD003 pre-adaptation to learn the other three dataset conditions. The training dataset contains labeled data from the source domain and unlabeled data from the target domain, and the test dataset contains unlabeled data from the rest of the target domain. In addition to the life label, the source domain dataset should also have a classification label. The classification label divides the entire lifecycle of the turbine engine into 10 segments, marked from 1 to 10, corresponding to 10 categories.

Table 1. C-MAPSS dataset.

Data	FD001	FD002	FD003	FD004
Engines: Training (N)	100	260	100	249
Engines: Testing	100	259	100	248
Operating Conditions	1	6	1	6
Faults Modes	1	1	2	2

Table 2. Dataset Partitioning.

Migration Process	Training Dataset		Test Dataset
FD004→ FD001	train_FD004	90% test_FD001 (no labeled)	10% test_FD001 (no labeled)
FD004→ FD002	(labeled)	90% test_FD002 (no labeled)	10% test_FD002 (no labeled)
FD003→ FD001	train_FD003	90% test_FD001 (no labeled)	10% test_FD001 (no labeled)
FD003→ FD002	(labeled)	90% test_FD002 (no labeled)	10% test_FD002 (no labeled))

3.2. Data Preprocessing

The four datasets are composed of three operation settings and 21 sensors. The operation setting parameters of FD001 and FD002 are shown in Figure 9. As the order of magnitude and dimension of the monitoring data of the 21 sensors are quite different, they need to be normalized before they are applied to the model.

$$x_i' = \frac{x_i - \mu_i}{\sigma_i}. \tag{16}$$

In Formula (16), μ_i and σ_i are distributed as the mean and standard deviation of the i-th characteristic signal x_i in x.

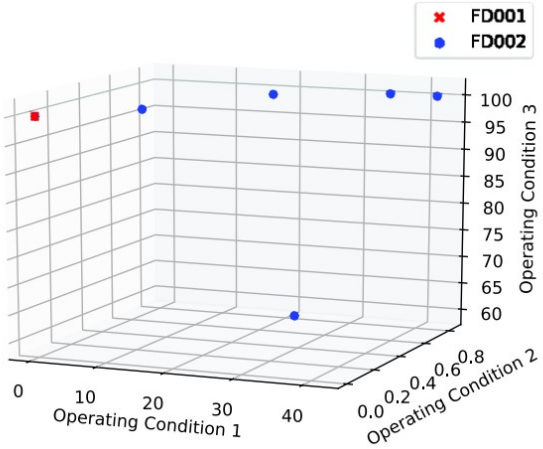

Figure 9. FD001, FD002 operating condition.

The normalized 21 sensor parameters for the four datasets are shown in Figure 10. As shown in this figure, the sensor distribution of FD001 and FD003 is similar, and the sensor distribution of FD002 and FD004 is also similar. This is because the above dataset pairs are generated under the same operating conditions, and the sensor values maintain a similar distribution before failure. However, due to different failure modes, the sensor data distribution shifts to a certain extent, which affects the RUL prediction.

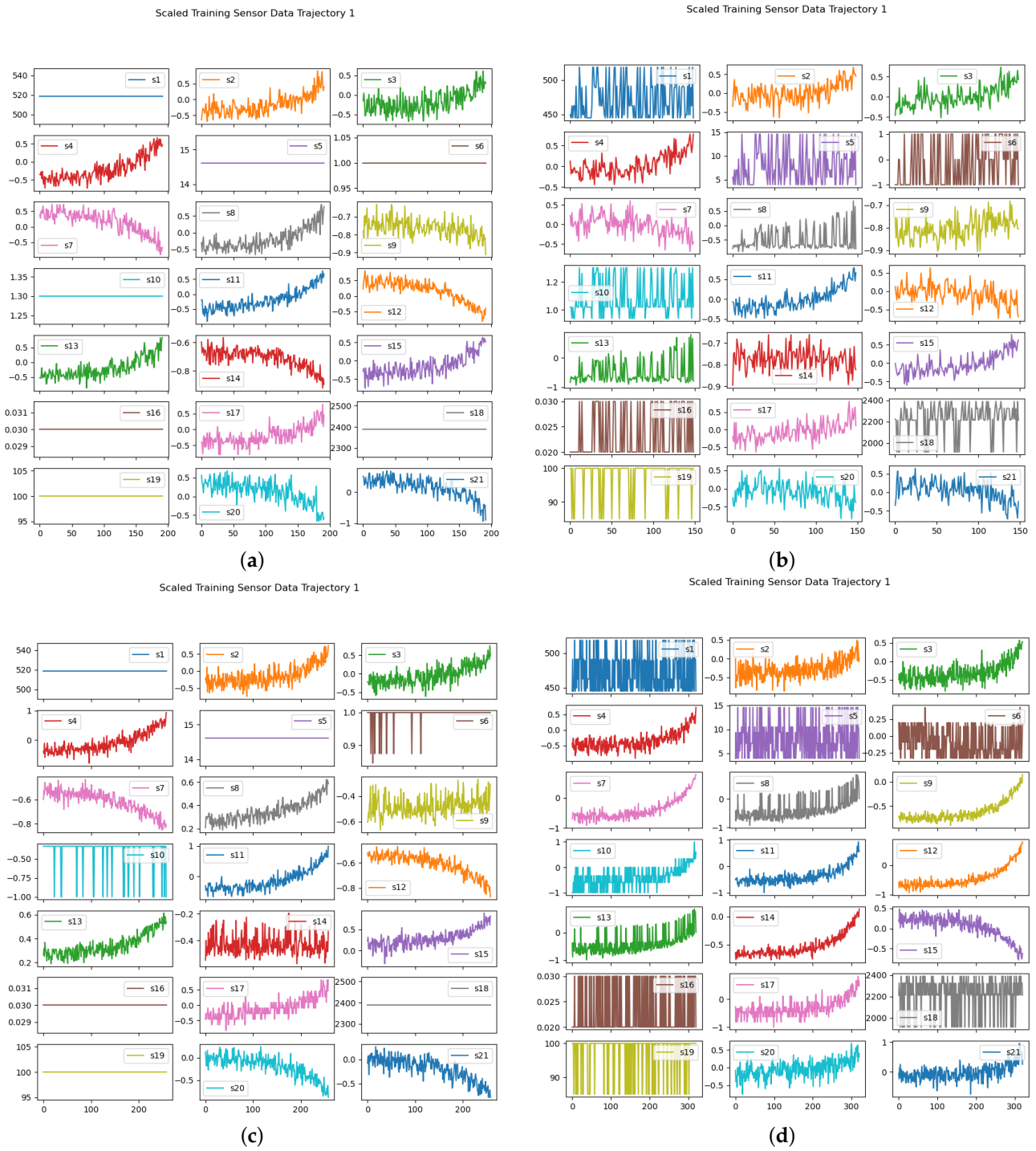

Figure 10. One-time degradation of full-cycle parameters for 21 sensors in four subsets. (**a**) FD001. (**b**) FD002. (**c**) FD003. (**d**) FD004.

Henme et al. [30] suggested that it is reasonable to estimate RUL as a constant value when the engine is operating under normal conditions. In our experiment, we also use the

method of segmented life to divide the decline of engine life into two processes. The decline process of the engine is relatively stable in the early stage of operation. After running for a period of time, the decline of engine performance is intensified. The maximum RUL of the engine is set to 125 cycles.

During training, the data monitored by the sensor is $X = \{X_1, X_2, X_3, \ldots, X_i, \ldots, X_n\} \in R^{m \times n}$, where m is the number of sensors and the size is 21, and n is the length of the time series. When the equipment runs to the i-th cycle, the measured value of the corresponding m-dimensional sensor is expressed as $X_i = \{x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,j}, \ldots, x_{i,m}\} \in R^{m \times 1}$. As the sensor data is a multivariate time series, we transform the source and target domain datasets using a sliding window with T_W = 30, time step $t_d = 1$, and take it as a mini-batch sample of the input data. A mini-batch of samples can be represented as:

$$\begin{aligned} X_{i:i+T_W-1} &= \{X_i, X_{i+1}, X_{i+2}, \ldots, X_{i+T_W-1}\} \\ &= \begin{bmatrix} x_{i,1} & x_{i,2} & \cdots & x_{i,m} \\ x_{i+1,1} & x_{i+1,2} & \cdots & x_{i+1,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i+T_W-1,1} & x_{i+T_W-1,2} & \cdots & x_{i+T_W-1,m} \end{bmatrix} \\ &\in R^{m \times T_W}. \end{aligned} \tag{17}$$

3.3. Life Prediction Process and Results

In our experiment, the feature extractor and RUL predictor are pre-trained using only the source domain dataset and labels, the backpropagation algorithm is used to update the model parameters, and the stochastic gradient descent optimization algorithm (SGD) is used to iteratively minimize the overall error. LSTM-MRAN network structure is shown in Table 3: Adam optimizer is used for 100 epochs, the learning rate is 0.001, and 32 mini-batch samples are input for each training. MRAN training follows the standard mini-batch SGD. The trained parameters are used as initialization parameters, and the target domain samples are input to obtain virtual classification labels. Then, the mini-batches of the source and target domains are simultaneously fed into the domain adaptation model. To eliminate the bias caused by the domain size, we oversample the domains with fewer datasets to ensure that the same number of source and target domains are input with each time data. The learning rate of the domain adaptation module is set to 0.001, and each training takes 32 samples for 50 epochs. After the model training is complete, RUL prediction can be performed.

Table 3. LSTM-MRAN network parameters.

Parameter	Value
T_W	30
LSTM layers	3
LSTM units	100
LSTM activation function	relu
LSTM dropout	0.3
MRAN substructure layers	3
Number of convolution kernels	32
Convolution kernel length	7/12/17
Substructure activation function	Tanh
λ	0.8
Learning rate	0.1
Batches	512
Optimizer	Adam
fully connected layers	3
fully connected units	(32, 16, 1)

In the MRAN structure, the size of the convolution kernel of each substructure has an impact on the prediction results. The MRAN in this study uses three substructures and uses three convolution kernels of different scales, respectively. A large number of experiments show that the prediction results are better when the convolution kernel size is 12. Therefore, to select better substructure convolution kernels, we take 12 as the center and d as the tolerance and use the FD001 dataset for comparative experiments. As shown in Figure 11, when the tolerance d = 5, the RMSE and R2-score values are the smallest. Therefore, in this experiment, the convolution kernels of the three substructures we choose are 7, 12, and 17.

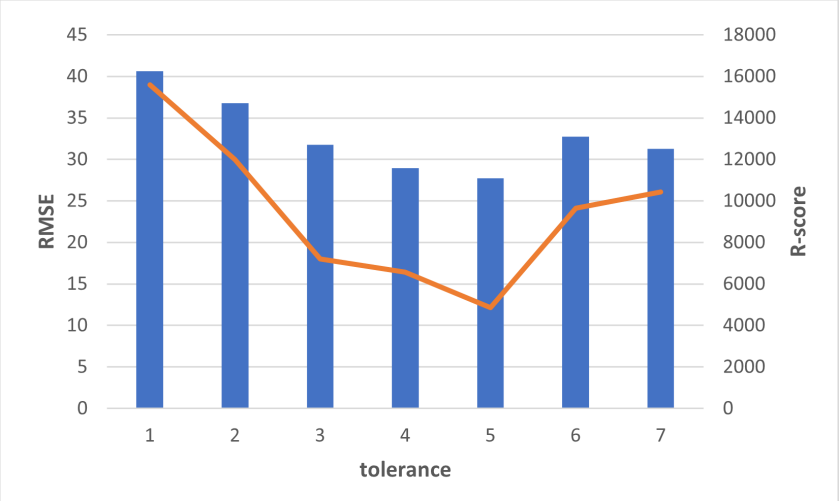

Figure 11. The effect of convolution kernel size on prediction results.

The processed test data sets are compared between the model without domain adaptation and the model that has undergone domain adaptation. The training results are shown in Figures 12 and 13. In these figures, (a) and (c) are the results of directly using the pre-trained model to perform RUL prediction on the target domain, (b) and (d) are the RUL prediction results after domain adaptation using the MRAN model. To better quantify the quality of the prediction results of the model RUL, we use root mean square error (RMSE) and score function s as the evaluation index of the prediction results:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2} \tag{18}$$

$$\begin{aligned} s &= \sum_{i=1}^{N} s_i \\ s_i &= \begin{cases} e^{-\frac{d_i}{13}} - 1, & \text{for } d_i < 0 \\ e^{\frac{d_i}{10}} - 1, & \text{for } d_i \geq 0, \end{cases} \end{aligned} \tag{19}$$

where $d_i = \hat{y}_i - y_i$, y_i and $\hat{y}_i$ are the theoretical value and predicted value of RUL.

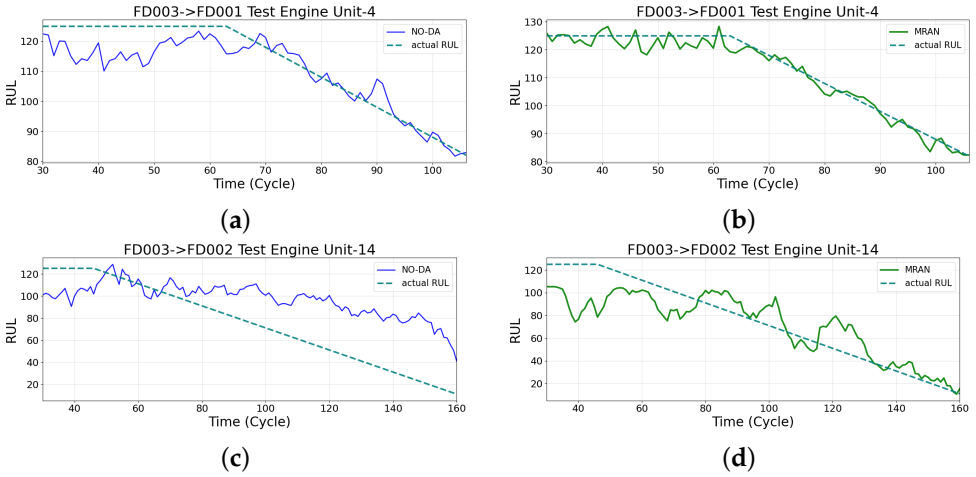

Figure 12. The source domain is FD003, and (**a**) is the prediction result of FD001 without domain adaptation. (**b**) FD001 prediction result after domain adaptation. (**c**) Prediction results of FD002 without domain adaptation. (**d**) FD002 prediction result after domain adaptation.

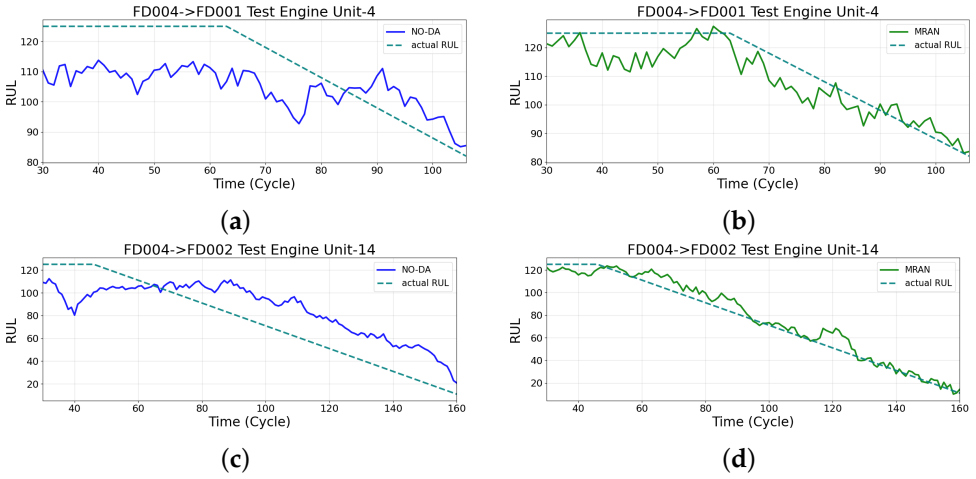

Figure 13. The source domain is FD004, and (**a**) is the prediction result of FD001 without domain adaptation. (**b**) FD001 prediction result after domain adaptation. (**c**) Prediction results of FD002 without domain adaptation. (**d**) FD002 prediction result after domain adaptation.

In the experiments, we found that methods using domain adaptive learning predict significantly better results than those without adaptive learning. This result proves that the method using transfer learning can learn cross-domain invariant features under different working conditions and show better results in RUL prediction. Comparing the target domain prediction results of FD001 and FD002 using FD003 and FD004 as the source domain, we can find that our method has better prediction results for FD001 in FD003 and better prediction results for FD002 in FD004. This is because the operating conditions of FD001 and FD003, FD002 and FD004 are similar, the sensor data degradation distribution is similar, and the transfer learning effect is better.

To verify the effectiveness of CMMD and Multi representation substructure in the MRAN model, we conducted five sets of experiments for comparison:

(1) Only the source domain is used to train the model, without domain adaptation
(2) Single representation domain adaptation (DAN)
(3) Multi-representation domain adaptation using MMD
(4) Single representation domain adaptation using CMMD
(5) Multi-representation domain adaptation using CMMD

The results are shown in Table 4. A comparison of the prediction results of (3), (4), and (5) can show that CMMD and multi-representation structure are better than MMD and the single-representation structure in the adaptive performance of the transfer learning field, and the two combined with better performance in domain adaptive learning.

Table 4. RMSE in different MRAN.

Source	Target	Source-only	DAN	MRAN (MMD)	DAN (CMMD)	MRAN (CMMD)
FD003	FD001	42.32 ± 5.61	41.33 ± 4.84	38.33 ± 3.74	39.68 ± 3.17	30.87 ± 2.57
	FD002	53.17 ± 6.87	47.68 ± 5.96	43.14 ± 5.31	45.32 ± 4.24	42.26 ± 3.36
Source	Target	Source-only	DAN	MRAN (MMD)	DAN (CMMD)	MRAN (CMMD)
FD004	FD001	45.21 ± 5.82	44.81 ± 5.53	38.33 ± 3.39	39.68 ± 3.16	32.87 ± 2.57
	FD002	47.60 ± 6.13	42.96 ± 5.97	32.57 ± 3.61	35.32 ± 3.35	27.32 ± 2.55

Comparison with Other Transfer Learning Methods

To prove the feasibility of the proposed LSTM-MRAN model in RUL prediction, it has been compared with other transfer learning methods such as JAN [31], DAN [26], DANN [32], CNN+MMD [20] and LSTM+DANN [22]. In [31], the paper proposed a deep learning method with joint adaptation networks. In [26], the paper introduced a deep adaptation network to learn transferable features. We use these two methods to conduct experiments. In [32], the paper designed a deep adversarial neural network to predict machinery RUL, which introduced adversarial training to achieve data alignments of different machine entities in order to extract generalized prognostic knowledge. The results are reported in Table 5. As shown in this table, the prediction performance of the MRAN model is better than that of other methods on the source domain FD004, FD003, and the target domain FD001 with large distribution differences, while the prediction performance of the FD002 dataset with similar distribution is similar to that of JAN. The reason is that MRAN aligns the extracted features in multiple spaces, which can more accurately extract the domain-invariant features between the source and target domains, thereby effectively improving the RUL prediction accuracy. This finding is significant for the RUL prediction of equipment with complex multi-dimensional fault characteristics.

Table 5. RMSE in different DA.

Source	Target	DAN	JAN	DANN	CNN+MMD [20]	LSTM+DANN [22]	Proposed method
FD003	FD001	39.2 ± 0.8	33.5 ± 0.2	36.9 ± 0.4	18.3 ± 1.2	31.74 ± 0.98	29.84 ± 2.73
	FD002	47.6 ± 5.3	45.4 ± 5.8	49.4 ± 6.7	48.8 ± 3.0	44.62 ± 1.21	42.23 ± 3.43
	FD004	62.5 ± 3.1	54.2 ± 3.5	55.3 ± 7.5	52.5 ± 2.7	47.96 ± 5.78	43.51 ± 3.64
Source	Target	DAN	JAN	DANN	CNN+MMD [20]	LSTM+DANN [22]	Proposed method
FD004	FD001	45.2 ± 5.6	36.5 ± 4.7	32.5 ± 3.4	35.3 ± 3.1	31.54 ± 2.42	32.87 ± 2.57
	FD002	31.4 ± 0.4	32.2 ± 0.1	33.2 ± 0.4	29.4 ± 0.6	29.73 ± 0.37	27.32 ± 2.55
	FD003	45.4 ± 3.9	44.2 ± 4.3	29.7 ± 3.8	38.7 ± 3.2	27.87 ± 2.69	36.05 ± 2.83

4. Conclusions

This study presents an RUL prediction method based on LSTM-MRAN, which adaptively aligns the source domain with the target domain through MRAN, and uses the LSTM model to fit the timing and nonlinear relationship of multi-sensor data in complex systems, providing an RUL prediction method under variable conditions. Compared with other methods used in the CMAPSS datasets, the effectiveness of this method for aircraft engine prediction has been verified, especially when the distribution of datasets in the source and target domains is quite different. Compared with other single-representation domain adaptive structures, our method has lower prediction loss and more accurate results, which shows that the domain invariant features extracted from multi-representation domain adaptive structures are more representative and can better reflect the degradation trend of target domain data. However, this structure has some limitations. First, the weight of

each substructure is consistent. However, in practice, the features extracted from each substructure may have different effects on RUL prediction. Second, it only implements domain adaptation between a single source domain and the target domain. In practical applications, source domain datasets may come from multiple different distributions. In future research, we point out that the attention mechanism is introduced to weigh the features extracted from different substructures. Moreover, we will further study the application of multi-source domain adaptation in the RUL prediction field.

Author Contributions: Methodology, Q.Z.; Project administration, Y.L.; Validation, Z.W.; Writing—original draft, Q.Z.; Writing—review & editing, Y.L. and A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Zhongshan Social Public Welfare Project 200824103628344.

Data Availability Statement: The data presented in this study are available from the corresponding author upon reasonable request.

Acknowledgments: We are grateful to the anonymous reviewers for their useful comments which helped us to improve the paper.

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

References

1. Atamuradov, V.; Medjaher, K.; Dersin, P.; Lamoureux, B.; Zerhouni, N. Prognostics and health management for maintenance practitioners-review, implementation and tools evaluation. *Int. J. Progn. Health Manag.* **2017**, *8*, 1–31. [CrossRef]
2. Zhu, J.; Nan, C.; Peng, W. Estimation of Bearing Remaining Useful Life based on Multiscale Convolutional Neural Network. *IEEE Trans. Ind. Electron.* **2018**, *66*, 3208–3216. [CrossRef]
3. Liu, J.; Saxena, A.; Goebel, K.; Saha, B.; Wang, W. *An Adaptive Recurrent Neural Network for Remaining Useful Life Prediction of Lithiumion Batteries*; NASA Ames Research Center: Mountain View, CA, USA, 2010.
4. Chen, X.; Xiao, H.; Guo, Y.; Kang, Q. A multivariate grey RBF hybrid model for residual useful life prediction of industrial equipment based on state data. *Int. J. Wirel. Mob. Comput.* **2016**, *10*, 90–96. [CrossRef]
5. Miao, J.; Li, X.; Ye, J. Predicting research of mechanical gyroscope life based on wavelet support vector. In Proceedings of the 2015 First International Conference on Reliability Systems Engineering (ICRSE), Beijing, China, 21–23 October 2016.
6. 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.
7. Li, X.; Zhang, W.; Ding, Q. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. *Reliab. Eng. Syst. Saf.* **2019**, *182*, 208–218. [CrossRef]
8. Malhi, A.; Yan, R.; Gao, R.X. Prognosis of Defect Propagation Based on Recurrent Neural Networks. *IEEE Trans. Instrum. Meas.* **2011**, *60*, 703–711. [CrossRef]
9. Wu, Y.; Yuan, M.; Dong, S.; Lin, L.; Liu, Y. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing* **2018**, *275*, 167–179. [CrossRef]
10. Zhang, J.; Wang, P.; Yan, R.; Gao, R.X. Long short-term memory for machine remaining life prediction. *J. Manuf. Syst.* **2018**, *48*, 78–86. [CrossRef]
11. 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]
12. Cao, Y.; Ding, Y.; Jia, M.; Tian, R. A novel temporal convolutional network with residual self-attention mechanism for remaining useful life prediction of rolling bearings. *Reliab. Eng. Syst. Saf.* **2021**, *215*, 107813. [CrossRef]
13. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [CrossRef]
14. Bousmalis, K.; Trigeorgis, G.; Silberman, N.; Krishnan, D.; Erhan, D. Domain separation networks. In *Advances in Neural Information Processing Systems 29 (NIPS 2016)*; Curran Associates, Inc.: Red Hook, NY, USA, 2016.
15. Ghifary, M.; Kleijn, W.B.; Zhang, M.; Balduzzi, D.; Li, W. Deep Reconstruction-Classification Networks for Unsupervised Domain Adaptation. *arXiv* **2016**, arXiv:1607.03516.
16. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-Adversarial Neural Networks. *arXiv* **2014**, arXiv:1412.4446.
17. Lu, W.; Liang, B.; Yu, C.; Meng, D.; Tao, Z. Deep Model Based Domain Adaptation for Fault Diagnosis. *IEEE Trans. Ind. Electron.* **2016**, *64*, 2296–2305. [CrossRef]
18. Li, X.; Zhang, W.; Ding, Q.; Sun, J.Q. Multi-Layer domain adaptation method for rolling bearing fault diagnosis. *Signal Process.* **2019**, *157*, 180–197. [CrossRef]
19. Han, T.; Liu, C.; Yang, W.; Jiang, D. A novel adversarial learning framework in deep convolutional neural network for intelligent diagnosis of mechanical faults. *Knowl.-Based Syst.* **2019**, *165*, 474–487. [CrossRef]

20. Zhang, W.; Li, X.; Ma, H.; Luo, Z.; Li, X. Transfer learning using deep representation regularization in remaining useful life prediction across operating conditions. *Reliab. Eng. Syst. Saf.* **2021**, *211*, 107556. [CrossRef]
21. Cao, Y.; Jia, M.; Ding, P.; Ding, Y. Transfer learning for remaining useful life prediction of multi-conditions bearings based on bidirectional-GRU network. *Measurement* **2021**, *178*, 109287. [CrossRef]
22. de Oliveira da Costa, P.R.; Akçay, A.; Zhang, Y.; Kaymak, U. Remaining useful lifetime prediction via deep domain adaptation. *Reliab. Eng. Syst. Saf.* **2020**, *195*, 106682. [CrossRef]
23. Fu, S.; Zhang, Y.; Lin, L.; Zhao, M.; Zhong, S.S. Deep residual LSTM with domain-invariance for remaining useful life prediction across domains. *Reliab. Eng. Syst. Saf.* **2021**, *216*, 108012. [CrossRef]
24. Ghifary, M.; Kleijn, W.; Zhang, M. Domain Adaptive Neural Networks for Object Recognition. *arXiv* **2014**, arXiv:abs/1409.6041.
25. Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep Domain Confusion: Maximizing for Domain Invariance. *arXiv* **2014**, arXiv:abs/1412.3474.
26. Long, M.; Wang, J. Learning Transferable Features with Deep Adaptation Networks. *arXiv* **2015**, arXiv:abs/1502.02791.
27. Zhu, Y.; Zhuang, F.; Wang, J.; Chen, J.; Shi, Z.; Wu, W.; He, Q. Multi-Representation Adaptation Network for Cross-Domain Image Classification. *arXiv* **2022**, arXiv:abs/2201.01002.
28. Ding, Y.; Jia, M.; Cao, Y. Remaining Useful Life Estimation under Multiple Operating Conditions via Deep Subdomain Adaptation. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 3516711. [CrossRef]
29. Saxena, A.; Goebel, K.; Simon, D.; Eklund, N. Damage propagation modeling for aircraft engine run-to-failure simulation. In Proceedings of the 2008 International Conference on Prognostics and Health Management, Denver, CO, USA, 6–9 October 2008; pp. 1–9. [CrossRef]
30. Heimes, F.O. Recurrent neural networks for remaining useful life estimation. In Proceedings of the 2008 International Conference on Prognostics and Health Management, Denver, CO, USA, 6–9 October 2008.
31. Long, M.; Zhu, H.; Wang, J.; Jordan, M.I. Deep transfer learning with joint adaptation networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2208–2217.
32. Li, X.; Zhang, W.; Ma, H.; Luo, Z.; Li, X. Data alignments in machinery remaining useful life prediction using deep adversarial neural networks. *Knowl.-Based Syst.* **2020**, *197*, 105843. [CrossRef]