Sound & Vibration 2024, 59(1), 1685.
https://doi.org/10.59400/sv.v59i1.1685

Article

A remaining useful life prediction method based on CNN-BiLSTM feature transfer in a high-noise environment

Zhao Jiang[1], Yanxia Zhao[1,2,*], Wei Yu[1,3,*]

[1] Zhejiang Yuexiu University, Shaoxing 312069, China
[2] Zhejiang Gongshang University, Hangzhou 310018, China
[3] Tianjin University, Tianjin 300350, China
*** Corresponding authors:** Yanxia Zhao, 20212015@zyufl.edu.cn; Wei Yu, weiyu@zyufl.edu.cn

CITATION

Jiang Z, Zhao Y, Yu W. A remaining useful life prediction method based on CNN-BiLSTM feature transfer in a high-noise environment. Sound & Vibration. 2024; 59(1): 1685. https://doi.org/10.59400/sv.v59i1.1685

ARTICLE INFO

Received: 4 September 2024
Accepted: 13 September 2024
Available online: 4 November 2024

Abstract: Prognosis and health management (PHM) is a comprehensive technique for fault detection, prediction, and health management. However, achieving accurate predictions of remaining useful life (RUL) under complex working conditions such as is still High-Noise Environment a challenge. Therefore, this paper proposes a feature transfer model based on Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory Neural Networks (BiLSTM) to predict RUL. In the feature extraction stage, On the basis of signal decomposition using local mean values，CNN is used to extract the degradation features. Secondly, the health factors are constructed by monotonicity and correlation to filter the features again. Thirdly, it uses BiLSTM to model the time series data in the RUL prediction stage. Then, it introduces the transfer learning algorithm to solve the problem of different data distribution due to the inconsistent working conditions of mechanical equipment data and estimates the confidence interval of the RUL by the Monte Carlo simulation technique. Finally, the effectiveness of our constructed framework via CNN-BiLSTM model on a publicly available degradation simulation dataset of turbine engines.

Keywords: high-noise environment; prognostic and health management (PHM); remaining useful life (RUL); transfer learning; convolutional neural networks (CNN); bidirectional long short-term memory (BiLSTM)

1. Introduction

Prognosis and health management (PHM) is a comprehensive fault detection, prediction, and health management technology that not only monitors and diagnoses faults but also anticipates the occurrence of faults in advance [1,2]. Enable the system to take safe fault-tolerant control means or maintenance and repair before failure, thus achieving autonomous safety and security and minimal cost loss [3–5]. As one of the key technologies of PHM, remaining useful life (RUL) prediction technology can predict the RUL of machinery and equipment in advance by analyzing the operational data of machinery and equipment collected by sensors. In order to prevent catastrophic events caused by sudden equipment failure, maintenance strategies must be developed based on the RUL prediction model [6,7]. Therefore, improving the accuracy of RUL prediction is of great practical value in reducing the risk of maintenance decisions [8]. RUL is the prediction of future failure events of a system based on currently obtained monitoring information, thus determining the effective time interval from the current moment to the moment of system failure [9,10]. However, the complex operating conditions, variable failure modes, and measurement noise or random disturbances in the detection data make the prediction results inevitably uncertain [11]. Therefore, the

quantification of uncertainty is a core issue in the field of RUL and is the key to ensuring that the equipment can operate safely and reliably for a long period of time [12].

RUL prediction is mainly using the historical monitoring data of mechanical equipment, and the RUL prediction of mechanical equipment is carried out based on the monitoring data. However, through the analysis of historical monitoring data of mechanical equipment, which can be found that historical monitoring data present the following features: with time series features, monitoring data are mostly collected from sensors, sensor data are characterized by changes over time, with temporal correlation and regularity [13]; High dimension, due to the complex internal structure of most mechanical equipment, most equipment exists with 20 or more dimensions of sensor state data [14]; more total data, the equipment sensors in the process of operation of machinery and equipment to collect real-time equipment status information data in seconds, the total amount of data saved is huge [15]; different data features distribution, machinery and equipment in the actual process of operation of variable operating conditions, the parameters between the equipment is not the same, which means that the same equipment at different times to obtain different monitoring data features distribution [16–18].

Although statistical data-driven and deep learning-based methods are widely used in RUL forecasting and have achieved many results, there is still no systematic and effective solution to the problem of quantifying forecast uncertainty [19]. This is mainly reflected in the insufficient ability of statistical data-driven methods to handle big data and the insufficient ability of deep learning methods to quantify prediction uncertainty [20]. For example, traditional machine learning prediction models assume that the data of different engines are from the same operating conditions and that the training and testing sets are independently and identically distributed. However, there are differences in the distribution of engine degradation data under different operating conditions, which leads to a dramatic decrease in the performance of traditional RUL prediction models [21,22]. In the past, such problems were mainly solved by fine-tuning the model parameters, but this requires a large number of labeled training samples, a prerequisite that is difficult to meet in practical application scenarios, so more effective methods are needed for predicting the RUL under different operating conditions. Transfer learning can transfer features from the source domain to the target domain and reduce the distribution differences between the source and target domains by learning features that are invariant across domains [23,24]. Zhang et al. [25] implemented the transfer of source domain information by sharing parameters. Shen et al. [26] used a model transfer strategy to build a neural prediction model to achieve RUL prediction under different working conditions. Mao et al. [27] used transfer component analysis to find the common features between different bearings to achieve RUL prediction by constructing a support vector machine-based prediction model. The device degradation process is a continuous change process with backward and forward dependence on time, and the current information must be processed while predicting future information [28]. For the prediction problem of RUL, scholars usually use long short-term memory (LSTM) models [29].

In this paper, we propose a transfer learning-based method for RUL to address the RUL uncertainty. Firstly, the common degradation features are extracted from the source domain by using the convolutional neural (CNN). Secondly, the extracted degradation features are re-screened by the monotonicity and correlation indicator. Since the bidirectional long short-term memory (BiLSTM) model is suitable for mining time-series information, the filtered features are input into the BiLSTM model. Finally, the confidence interval of the remaining lifetime is estimated by Monte Carlo's (MC) simulation.

The contributions of this research are as follows:

1) Compared with the point estimation of remaining life, the proposed method can effectively solve the problem that the uncertainty of model prediction results is difficult to measure, which is of great practical value for reducing the risk of maintenance decisions.
2) Introduces the transfer learning algorithm to solve the problem of different data distribution brought by the non-uniform working conditions of mechanical equipment data, and the transfer learning-based BiLSTM method is proposed to solve the problem of RUL prediction under multiple working conditions.

The remainder of this paper is organized as below. In section 2, the methods are briefly described. Section 3 introduces the RUL prediction model based on domain adaptive. In section 4, the RUL prediction based on CNN-BiLSTM is put forward. In section 5, by taking the turbofan engine as an example, the proposed CNN-BiLSTM based on the transfer learning framework is verified. The conclusions are given in section 6.

2. Methods

2.1. Local mean decomposition

Let the vibration signal obtained by monitoring be $x(t), t = 1,2\cdots, L$ ， t is the sampling time，L is the total sampling time. The steps to implement the LMD method are as follows:

(1) The mean m_i and envelope estimates χ_i of the extreme points are smoothed by applying the sliding average method to obtain the local mean function $m_{11}(t)$ and the envelope estimate function $\chi_{11}(t)$.

(2) The local mean function $m_{11}(t)$ is separated from the rolling bearing vibration signal $x(t)$ and demodulated to obtain a pure FM signal $s_{11}(t)$.

$$h_{11}(t) = x(t) - m_{11}(t) \tag{1}$$

$$s_{11}(t) = h_{11}(t)/\chi_{11}(t) \tag{2}$$

Repeat until the envelope function χ_i satisfies $\lim_{n\to\infty}\chi_{1n}(t) = 1$.

(3) Multiplying the envelope function of the above iterative process yields the envelope signal.

$$\chi_1(t) = \chi_{11}(t) \times \chi_{12}(t) \cdots \chi_{1n}(t) \tag{3}$$

(4) FM signal Multiplied with the envelope signal $\chi_1(t)$, Product Functions (PF) component.

$$PF_1(t) = s_{1n}(t) \times \chi_1(t) \tag{4}$$

(5) The new signal $\mu_1(t)$ is obtained by separating it from the vibrational signal $PF_1(t)$ and repeating it as a vibrational signal until the iteration stops as a monotonic function $\mu_q(t)$.

$$\begin{cases} \mu_1(t) = x(t) - PF_1(t) \\ \mu_2(t) = \mu_1(t) - PF_2(t) \\ \vdots \\ \mu_q(t) = \mu_{q-1}(t) - PF_q(t) \end{cases} \tag{5}$$

(6) The rolling bearing vibration signal is decomposed into a series of PF components and a residual component $\mu_q(t)$.

$$x(t) = \sum_{i=1}^{q} PF_i\,(t) + \mu_q(t) \tag{6}$$

2.2. Particle filter

In rolling bearing RUL prediction, it is often difficult to observe the state of degradation of rolling bearings [30]. In contrast, existing sensing technologies such as vibration sensors and acoustic emission sensors provide indirect measurements for the assessment of the degree of bearing degradation [31,32]. Based on the Bayesian framework, this scenario of indirectly reacting to the internal state of a device can be described by the following mathematical model.

$$x_k = f(x_{k-1}, \mu_{k-1}) \tag{7}$$

$$y_k = h(x_k, \vartheta_k) \tag{8}$$

where: k is the time; $f(\cdot)$and $h(\cdot)$is the transfer function of the system state values and observations; x_k and y_k is the system state and measurements at the moment k; $\mu_{k-1}$and ϑ_k is the white noise in the model.

Particle filtering is a Monte Carlo method based on lem of particle exhaustion. In this study, a regularized resampling method based on the Euclidean distance kernel function is designed based on the regularization idea. The method selects a number of steps of iterative update of particles before the current moment as the time scale, calculates the Euclidean distance between the observed value and the estimated value of each particle in the time scale, and then uses this distance as the basis for adjusting the weights of particles, which solves the problem that the original regularized resampling methods based on Epanechnikov kernel density function, Gaussian kernel function, etc., only pay attention to the current value of particles and ignore the updating trend. The implementation steps of the regularized particle filtering algorithm are as follows.

(1) For $i = 1,2,\cdots,N$, generate particles $\left\{x_0^{(i)}\right\}_{i=1}^{N}$ from the a priori distribution$p(x_0$〉. For $k = 1,2,\cdots,N$, perform step$2 \rightarrow 4$.

(2) Importance Sampling: Based on $p\left(x_k|x_{k-1}^{(i)}\right)$sampling N particles to form a collection of particles $\left\{x_k^{(i)}\right\}_{i=1}^{N}$.

(3) Weight Calculation and Normalization: Calculate the observation estimate y_k^i for each particle, and after the current observation value is input, calculate the weight $\omega_k^{(i)}$ of each sample in the sample set.

$$\omega_k^{(i)} = \frac{1}{\sqrt{2\pi}} e^{(-\frac{(y_k^i - y_k)^2}{2\sigma})} \tag{9}$$

Normalize the computation to generate a collection of particles$\left\{x_k^{(i)}, \omega_k^{(i)}\right\}_{i=1}^{N}$.

$$\omega_k^{(i)} = \frac{\omega_k^{(i)}}{\sum_{i=1}^{N} \omega_k^{(i)}} \tag{10}$$

(4) State estimation.

$$X_k = \sum_{i=1}^{N} x_k^{(i)} \omega_k^{(j)} \tag{11}$$

2.3. Dynamic modeling for RUL prediction

Mathematical modeling can be applied to different data domains. The time-series data $X_S = [x_{S1}, x_{S2}, \ldots, x_{Sn}] \in Z^{\Lambda \times n}$ from a labeled source domain under certain operating conditions and the unlabeled target domain data $X_T = [x_{T1}, x_{T2}, \ldots, x_{Tn}] \in Z^{\Lambda * n}$under another set of operating conditions, where Λ is the number of sensors and n is the number of data samples. A model is trained on the labeled source domain data, which then yields the RUL prediction value $\hat{y}_S = [\hat{y}_{S1}, \hat{y}_{S2}, \ldots, \hat{y}_{Sn}]$ for each instance in the source domain data.

In transfer learning, given a data sample $X = [x_1, x_2, \ldots, x_n]$ and its sample space χ, the data domain D is composed of the source domain data sample space and the target domain data sample space. Under different operating conditions, the sample feature spaces of the source domain D_S and the target domain D_T are different, leading to different edge probability distributions $p(X_S) \neq p(X_T)$, i.e., The transfer learning task is composed of the sample label space Y and the sample conditional probability distribution $p(Y|X)$. When the source domain sample label space is different, or the sample conditional probability $p(Y_S|X_S) \neq p(Y_T|X_T)$ is different under different operating conditions, since the prediction label types corresponding to the source domain data and the target domain data are consistent, but the sample conditional probability distribution is different, the aim is to transfer the RUL prediction model trained on the source domain data to the target domain, i.e., the function F makes p satisfy the following formula:

$$p(F(X_S)) = p(F(X_T)) \tag{12}$$

$$p(X_S|F(X_S)) = p(X_T|F(X_T)) \tag{13}$$

Thus, the model trained on the source domain data can effectively predict the RUL (Remaining Useful Life) of the target domain data, yielding the predicted RUL value $\hat{y}_T = [\hat{y}_{T1}, \hat{y}_{T2}, \ldots, \hat{y}_{Tn}]$ for the target domain.

2.4. Convolutional neural network

The CNN is essentially an input-to-output mapping, which does not require any precise mathematical expression between input and output as long as the CNN is trained and has the ability to map between input and output pairs. A typical CNN mainly consists of an input layer, a convolutional layer, a down sampling layer (pooling layer), a fully connected layer, and an output layer. The composition structure is shown in **Figure 1**.

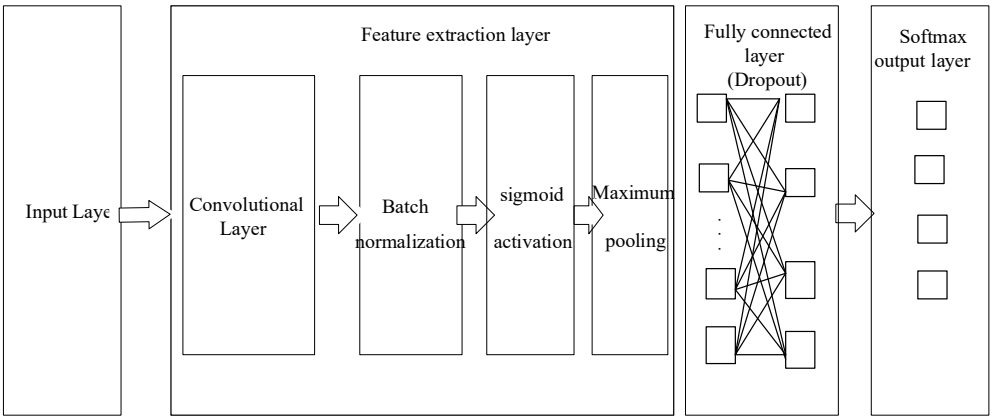

Figure 1. Composition of CNN.

(1) Convolutional Layer

By using multiple convolution kernels to perform convolution operations with the input matrix, feature vectors are obtained through an activation function. The mathematical expression for this is as follows:

$$X_j^l = f\left(\sum_{i \in M_j} X_i^{l-1} \times w_{ij}^l + b_j^l\right) \tag{14}$$

where X_j^i represents the jth element in the lth layer; $M_j$the jth convolutional region in the lth layer. w_{ij}^l represents the weight matrix of the corresponding convolutional kernel. b_j^l represents the bias term. $f(\cdot)$denotes the activation function, commonly using the ReLU function.

(2) Pooling Layer

After the input data goes through feature extraction with convolutional kernels, due to the large size of the data, it is often followed by a pooling layer to reduce the number of model weight parameters, improve computational speed, and mitigate overfitting issues. The calculation formula for the pooling layer is as follows:

$$X_j^l = f(\beta_j^l down(X_j^{l-1}) + b_j^l) \tag{15}$$

where β represents weight matrix ,$down(\cdot)$represents the down sampling function.

Pooling can be mainly divided into two types: average pooling and max pooling. Among them, max pooling is the most commonly used method, and its expression is as follows:

$$X^l = \max_{(i-1)l+1 \le t \le il} X^{l-1}(t) \tag{16}$$

where l represents the length of the pooling region.

(3) Fully Connected Layer

After the input data has been alternated with multiple convolutions and pooling, the extracted features are classified through a fully connected layer, which contains multiple implicit layers to improve the generalization performance of the model.

(4) Dropout layer

The dropout regularization technique ignores neurons in a certain proportion, added before the fully connected layer, which can prevent the model from overfitting the phenomenon. The standard process is shown by the following formula.

$$y = f(Wx) \times m, m_i \sim Bernoulli(n, p) \tag{17}$$

where x is the input, W is the input weights, and y is the output.

In this paper, we first use the degradation features extracted by CNN for re-screening to construct HI that can reflect the degree of deviation from the health state due to degradation.

$$HI = \sqrt{\sum_{t=1}^{q} (f_t - f_{health})^2} \tag{18}$$

where f_t are the real-time degradation feature f_{health} and the initial health feature, and q is the length of the sequence of HI. Then, the similarity between the depth features of the target domain and the HI sequence is calculated to extract the common features for the purpose of degraded information migration. Specifically, Dynamic Time Warping (DTW) distance is used to measure the similarity. Different from traditional metrics, DTW can effectively solve the graph translation problem and is suitable for measuring the shape similarity of unequal sequences.

The specific steps of the algorithm are as follows:

(1) Calculate the DTW distance between the feature matrix F_i and HI to obtain the similarity matrix $Q_i = [q_1, q_2, \cdots q_L]^T$, where is the L -th dimension of the extracted features.

(2) Construct the weight matrix $\omega = [\omega_1, \omega_2, \cdots, \omega_n]$, which is calculated as follows: $\omega = \sum_{i=1}^{n} L - Index_i$. Where $Index_i$ denotes the similarity matrix in ascending order, and the higher the similarity, the higher the corresponding weight.

(3) Perform similarity ranking on the weight matrix to obtain the importance ranking of each feature.

2.5. Construction of health indicators

In order to effectively evaluate the HI constructed in this paper, correlation, and monotonicity indicators are used to measure it.

(a) Correlation

The correlation portrays the degree of correlation between the features and the time series. The greater the correlation between the two, the better the corresponding feature describes the engine. In this paper, the correlation between the feature and the time series is noted.

$$corr = \frac{\left|\sum_{t=1}^{T}(F_t - \tilde{F})(l_t - \tilde{l})\right|}{\sqrt{\sum_{t=1}^{T}(F_t - \tilde{F})^2 \sum_{t=1}^{T}(l_t - \tilde{l})^2}} \tag{19}$$

where, F_t and l_t are the t eigenvalues and corresponding moments of the first sample, respectively; $\tilde{F}$ and $\tilde{l}$ are the mean values of the sample eigenvalue series and time series, respectively; T is the number of samples in the whole life cycle. The correlation indicator is between 0 and 1, and the better the correlation between features and time, the closer the value is to 1.

(b) Monotonicity

With the degradation of engine performance, the degree of failure will become more and more serious, and the corresponding feature values will also show a degradation trend. In this paper, the monotonicity of the features and time series is as follows:

$$Mon = \left|\frac{num(dF > 0)}{T-1} - \frac{num(dF < 0)}{T-1}\right| \tag{20}$$

where $num(dF > 0)$ is the number of feature sequence difference values greater than zero; $num(dF < 0)$ is the number of feature sequence difference values less than zero; T is the length of the feature sequence. The monotonicity indicator is between 0 and 1, and the better the monotonicity of the features, the closer to 1. Otherwise, the closer to 0.

When making feature selection, correlation, and monotonicity need to be considered together. Therefore, in this paper, the above two indexes are linearly combined, and the comprehensive evaluation criterion is noted.

$$Cri = \omega_1 Corr + \omega_2 Mon \tag{21}$$

where ω_1 and ω_2 are the weights of the two weighting factors.

2.6. Time-series prediction model based on BiLSTM

BiLSTM is an improved model of LSTM. BiLSTM can consider the influence of time series in both past and future directions on the data of the current moment, which better captures the long-term dependence of time series [33].

The mathematical expression of LSTM.

$$f_t = o(w_f \times [h_{t-1}, x_t] + b_f) \tag{22}$$

$$i_t = o(w_i \times [h_{t-1}, x_t] + b_i) \tag{23}$$

$$\tilde{C}_t = \varphi\ (w_C \times [h_{t-1}, x_t] + b_C) \tag{24}$$

$$C_t = f_t \times C_{t-1} + i_t \tilde{C}_t \tag{25}$$

$$o_t = o(w_o \times [h_{t-1}, x_t] + b_o) \tag{26}$$

$$h_t = o_t \times \varphi(C_t) \tag{27}$$

where w_f, w_i, w_o, and$w_c$represent the weight matrices of the forgetting gate, input gate, output gate, and cell state, b_f、b_i、b_o and b_c represent the bias, σ and ϕ represent the sigmoid activation function and the activation function, respectively. h_{t-1} is the output state at the previous moment, x_t is the input at the current moment, f_t is the forgetting gate, i_t is the input gate, o_t is the output gate, $\tilde{c}_t$ is the candidate state, c_t is the memory cell, and c_{t-1} is the cell state at the moment.

The output $\vec{h}_t$ of the forward LSTM layer and the output $\overleftarrow{h}_t$ of the backward LSTM layer are calculated separately, and then the output is obtained by concatenating $\vec{h}_t$ and $\overleftarrow{h}_t$.

The mathematical expression of BiLSTM:

$$\vec{h}_t = LSTM(x_t, \vec{h}_{t-1}) \tag{28}$$

$$\overleftarrow{h}_t = LSTM(x_t, \overleftarrow{h}_{t+1}) \tag{29}$$

$$y_t = w_{\vec{h}y}\vec{h}_t + w_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y \tag{30}$$

where $w_{\vec{h}y}$ denotes the connection weight from the forward LSTM layer to the output layer, $w_{\overleftarrow{h}y}$the connection weight from the backward LSTM layer to the output layer, and the output layer's bias.

3. RUL prediction model based on domain adaptive

3.1. Domain adaptive framework

An adaptive layer is introduced in the deep neural-based prediction model to develop the RUL model adapted to the requirements of the target class. The adaptive domain framework is shown in **Figure 2**.

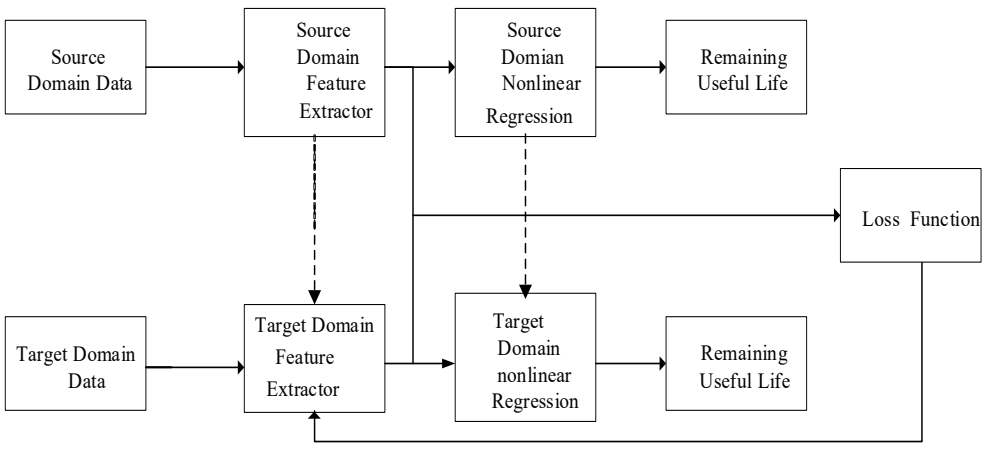

Figure 2. Domain adaptive framework.

(a) The RUL prediction model is trained using the source domain data, i.e., the process monitoring signals with historical performance degradation patterns, and the Adam optimizer is used to train the loss function with the mean absolute error.

$$\ell_{TRAIN} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{m} \tag{31}$$

where y_i is the theoretical value of remaining life, $\hat{y}$ is the predicted value of remaining life, m and is the training sample size of each batch.

(b) The target domain model in the initial state shares the parameters of the source domain model. The adaptive layer is added after the model feature extractor, and the regression loss function is used to measure the distance between the features extracted by the feature extractor in the source and target domains. The loss functions of the source domain features $D_S = \{x_i\}, x \in R^d$ and the target domain features $D_T = \{u_i\}, u \in R^d$:

$$\ell_{Loss} = \frac{1}{N} \sum_{i=1}^{N} |x_i - \mu_i| \tag{32}$$

During the model update, minimizing the source domain training loss itself is likely to lead to overfitting. Therefore, combining the source domain training loss and the loss between the two domains on the target domain model is trained jointly to update the parameters of the target domain feature extractor.

$$\ell = \ell_{TRAIN} + \sum_{i=1}^{t} \lambda_i \, \ell_{Loss} \tag{33}$$

where t denotes the number of loss layers in the deep; λ_i denotes the weight value that weighs the adaptive and training accuracy.

3.2. Pre-training and fine-tuning methods

If the target category data contains a small amount of labeled data, the RUL prediction model is adjusted by the labeled data within the target category through a pre-training-fine-tuning method. The pre-training-fine-tuning method refers to finding shared model parameter information from the source and target domains for the purpose of information transfer. The process of the pre-training-fine-tuning method for tuning the RUL prediction model is shown in **Figure 3**. In the process of fine-tuning the parameters of the LSTM, the nonlinear regressor in the model is fine-tuned using the process monitoring signal with labels in the target domain as input. The fine-tuning process is trained using the Adam optimizer with Equation (33) as the loss function.

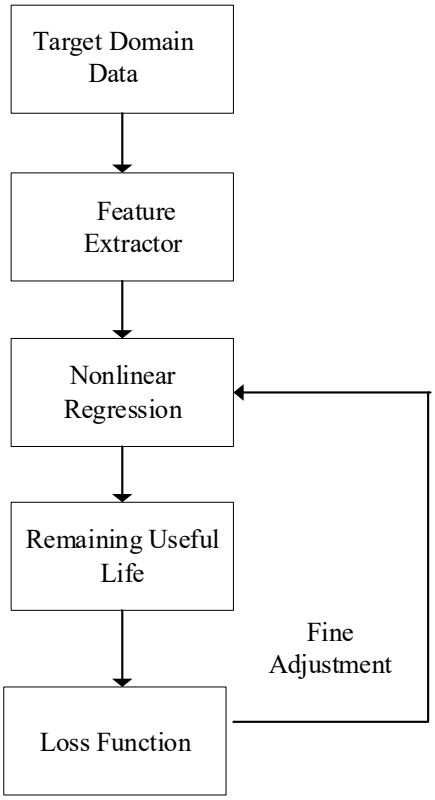

Figure 3. Pre-training and fine-tuning method.

4. RUL prediction model based on transfer learning

4.1. Process of feature transfer learning

In order to learn the common features between the source domain's fault data and preserve the unique features of the target domain, this paper proposes the CNN-BiLSTM prediction model based on transfer learning. The process of feature transfer learning is shown in **Figure 4**.

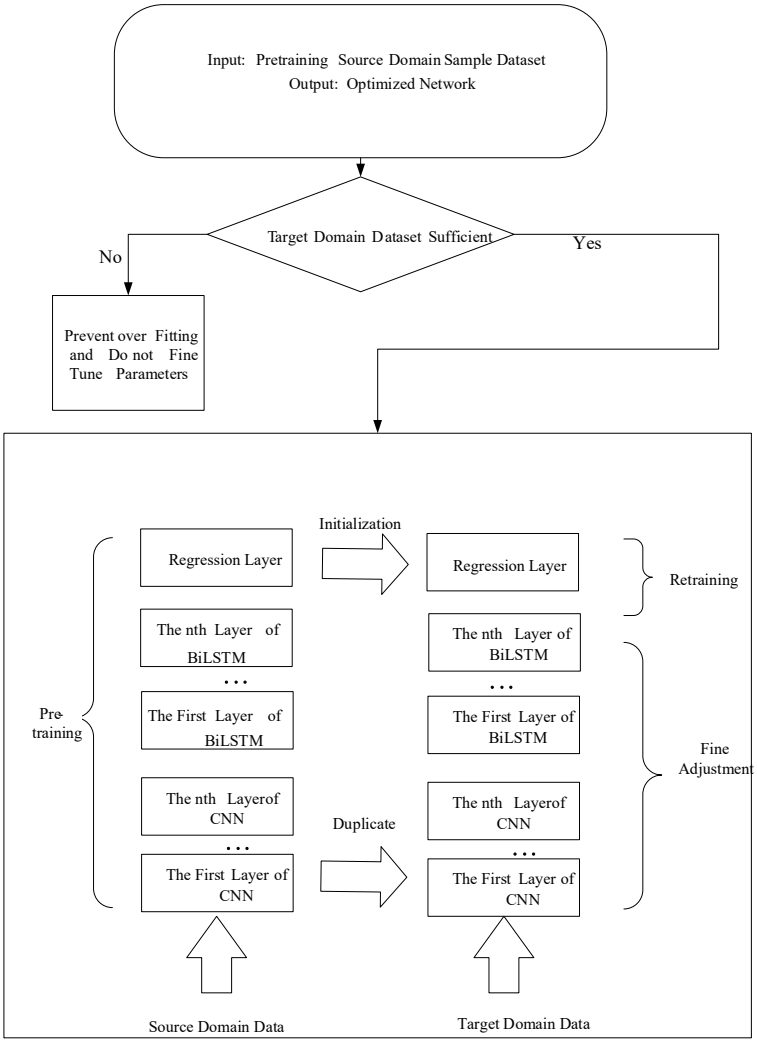

Figure 4. Schematic diagram of CNN-BiLSTM structure.

In **Figure 4**, this paper updates the model parameters by sharing the implicit layer. Parameter transfer, also known as model transfer, is one of the most intuitive and commonly used methods in transfer learning. The basic idea is to directly transfer the model parameters that have been pre-trained on a large-scale dataset to a new target task, and then fine-tune them according to the characteristics of the target task. Specific operations are as follows:

1) Load the pre-trained model: First, we need to load the pre-trained model.
2) Modify the output layer: Replace the last layer of the pre-trained model (usually a fully connected layer) with a new layer that matches the cat and dog classification task.
3) Freeze and fine-tune: Typically, we first freeze most of the layers of the pre-trained model (these layers are responsible for extracting the basic features of the image), and then only train the new classification layer. During the training process, gradually unfreeze some layers for fine-tuning as needed.

The parameters from the input layer to the hidden layer ensure that the target domain can learn features similar to the source domain while retaining features unique to the target domain.

4.2. Measure the uncertainty of the RUL model

Assuming a normal prior distribution of probabilities about the random variables ω in the BiLSTM and using Bayesian theory to determine the posterior distribution. The expression of the posterior distribution of the random variables ω.

$$p(\omega|X,Y) = \frac{p(Y|X,\omega)p(\omega)}{P(Y|X)} \tag{34}$$

where X and Y correspond to the training set consisting of input and output samples, respectively.

For a new input x^*, the predicted distribution is obtained from Equation (34).

$$p(y^*|x^*,X,Y) = \int p\,(y^*|x^*,\omega)p(\omega|X,Y)d\omega \tag{35}$$

The posterior distribution of the random variable is difficult to obtain directly, which is necessary to construct an approximate distribution $q^*(\omega)$ based on the variational inference to approximate and minimize the KL dispersion $KL(q(\omega)||p(\omega|X,Y))$ to determine.

$$p(y^*|x^*,X,Y) \approx \int p(y^*|x^*,\omega)q^*(\omega)d\omega \tag{36}$$

From Equation (36), to obtain the Bayesian posterior distribution, it is necessary to minimize the KL scatter, and minimizing the KL scatter is equivalent to maximizing the lower bound of the marginal likelihood function.

$$\begin{aligned} KL(q(\omega)||p(\omega|X,Y)) &= \int q(\omega)\,log\frac{q(\omega)}{p(\omega|X,Y)}d\omega \\ &= -\sum_{n=1}^{N}\int q(\omega)\,log(\,y_n|x_n,\omega)d\omega + KL(\,q(\omega)||p(\omega)) \end{aligned} \tag{37}$$

KL scatter regularization to obtain the loss function:

$$\ell = -\sum_{n=1}^{N} log\,p\,(y_n\,|x_n,\widehat{\omega}_n) + \sum_{i=1}^{L}(\frac{p_i c^2}{2}\left\|\mu_\omega^l\right\|_2^2 + \frac{c^2}{2}\left\|\mu_b^l\right\|_2^2) \tag{38}$$

where c is the length coefficient a priori $\widehat{\omega}_n \sim q(\omega)$, $p_i \in [0,1]$ denotes the probability set in advance, μ_ω^l and μ_b^l is the variational parameter of the weight matrix ω_i and the bias vector b_i, respectively.

Set ξ denote the weight decay parameter and obtain the loss function of the neural model using dropout as follows.

$$\ell \propto \ell_{dropout} = \frac{1}{N}\sum_{n=1}^{N}\ell(y_n,\hat{y}_n) + \xi\sum_{i=1}^{L}(\|\omega_i\|_2^2 + \|b_i\|_2^2) \tag{39}$$

Prediction uncertainty can be quantified by adding a dropout with probability $p$and a regular term with a weight recession coefficient ξ in the fully connected layer. The resulting loss function ℓ is equivalent to $\ell_{dropout}$. The loss function ℓ can be obtained by the gradient descent method to obtain the optimal weight matrix ω_i^*.

Given a new input x^*, MC performs V times to obtain the predicted mean and variance, i.e.,

$$p(o^*|x^*,X,O) \approx \int p(o^*|x^*,X,\omega)\, q(\omega) d\omega \approx \frac{1}{V}\sum_{v=1}^{V} p\ (o^*|x^*,\widehat{\omega}_v) \tag{40}$$

where, $\widehat{\omega}_v \sim q(\omega)$, O is the training data set.

4.3. Flow of RUL prediction model based on transfer learning

In this paper, propose a framework including feature extraction and RUL prediction, which consists of two main parts: the first part uses the CNN to extract degradation features and then filter them by HI; the second part uses the BiLSTM to mine timing information to predict the RUL of the engine. The flow of the proposed RUL prediction framework is shown in **Figure 5**. we propose a transfer learning-based method for RUL to address the RUL uncertainty. Firstly, the common degradation features are extracted from the source domain by using the convolutional neural (CNN). Secondly, the extracted degradation features are re-screened by the monotonicity and correlation indicator. Since the bidirectional long short-term memory (BiLSTM) model is suitable for mining time-series information, the filtered features are input into the BiLSTM model. Finally, the confidence interval of the remaining lifetime is estimated by Monte Carlo's (MC) simulation.

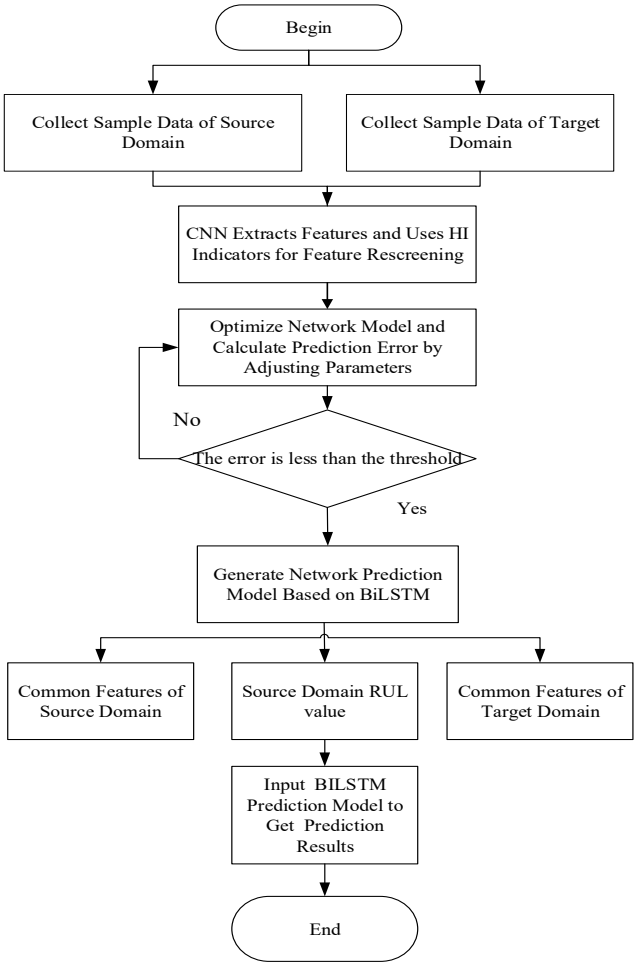

Figure 5. Flow chart of RUL prediction based on CNN-BiLSTM.

The specific steps are as follows:

Step 1: Acquisition and pre-processing of raw data. The data collected by multiple sensors are normalized as the original data of the source and target domains.

Step 2: The normalized data set is input into the CNN, and the extracted common degradation features are obtained by adjusting the hyperparameters of the model to minimize the reconstruction error.

Step 3: Construction of HI and re-screening of degenerate features. Consider the monotonicity and correlation between the features and the Health indicator.

Step 4: Select the BiLSTM as a decoder and introduce dropout into the BiLSTM model to generate the BiLSTM-based prediction model.

Step 5: Use the prediction error to optimize the BiLSTM model parameters.

Step 6: Input the source and target domain data into the BiLSTM-based prediction model to obtain the prediction results and measure the uncertainty of the RUL prediction results by the MC simulation technique.

5. Experimental verification

5.1. Dataset description

Because of the complex structure of aero engines, degradation data often present high dimensionality and large data volume [34]. In this paper, we take the CMAPSS dataset provided by NASA as an example, which contains four sub-data (FD001-FD004) of engines in different operating states and failure modes, and each sub-dataset contains full-life data of multiple engines sampled by 21 sensors under different operating conditions and divided into training and test sets, with specific information shown in **Table 1**. The C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset is published by the National Aeronautics and Space Administration (NASA) and is specifically designed for predicting the RUL of aircraft engines. The data is generated by simulating the operation of large commercial turbofan engines, encompassing sensor data under various working conditions and fault modes. The C-MAPSS dataset includes four sub-datasets (FD001, FD002, FD003, FD004), each containing different numbers of working conditions and fault states. These datasets are widely used in the fields of machine learning and data mining to support the health management systems of aircraft engines. The parameters in the dataset include operational parameters of the engine, such as vibration and temperature information, with a total of 24 sensor data points, which can be used to train and test fault diagnostic and RUL prediction models. The purpose of the experiment is to train a model based on the test data to achieve the estimation of the remaining running time of the test set data.

Perform an adaptive LMD (Local Mean Decomposition) decomposition on the accelerated life data of the aforementioned rolling bearing, and the results are shown in **Figure 6**.

Table 1. Aero engine dataset.

Number	Training Set	Testing Set	Operating Condition	Failure Mode
FD001	100	100	1	1
FD002	260	259	6	1
FD003	100	100	1	2
FD004	249	248	6	2

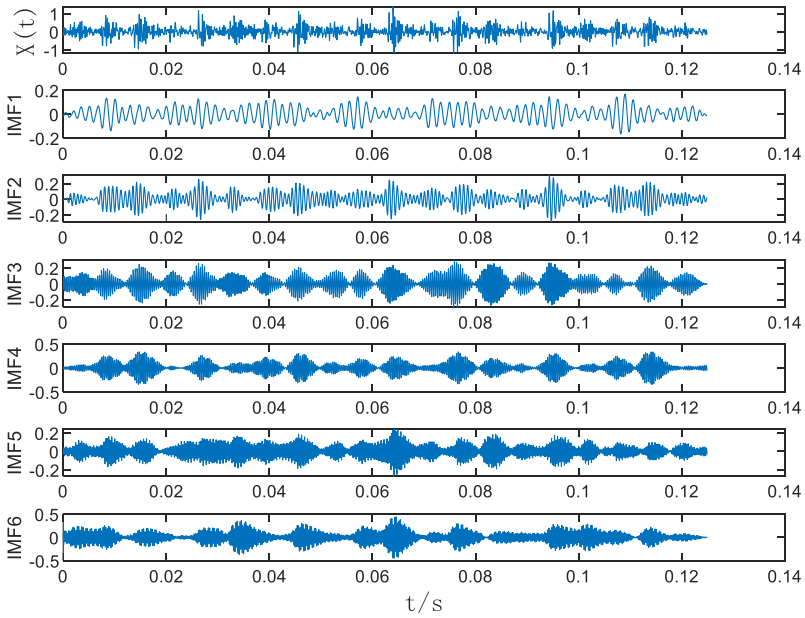

Figure 6. LMD decomposition results.

5.2. Parameter setting

The performance of the constructed model is influenced by the setting of the model parameters, and in order to measure the performance of the proposed prediction model and to improve the efficiency of parameter adjustment, the score function (SF) and the root mean square error (RMSE) are usually used to evaluate the prediction effect.

The formula for calculating SF:

$$SF = \begin{cases} e^{-\frac{h_i}{13}} - 1, & h_i < 0 \\ e^{\frac{h_i}{13}} - 1, & h_i \geq 0 \end{cases} \tag{41}$$

where, $h_i = \widehat{RUL}_i - RUL_i$, $\widehat{RUL}_i$is the predicted value of RUL and RUL_i is the true value of RUL.

The forecast results are measured using RMSE, which is calculated as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} h_i^2} \tag{42}$$

Smaller values of SF and RMSE mean better prediction accuracy.

The sliding time window is set to 30, and the number of iterations is set to 500. the structural parameters of the BiLSTM are set as shown in **Table 2**, and the model training process is shown in **Figure 7**.

Table 2. Parameters of BiLSTM.

BiLSTM	**Parameter**
Number of hidden layers	3
Network structure	50–50–50
Optimizer	Adam
Activation function Loss function Dropout	sigmoid-sigmoid-tanh Mean Square Error $\{0.2,\ 0.5\}$

5.3. Model training and result analysis

The test data contains 100 incomplete sequences, and the RUL of the corresponding engine is at the end of each sequence. Sort the training data by sequence length, as shown in **Figure 7**.

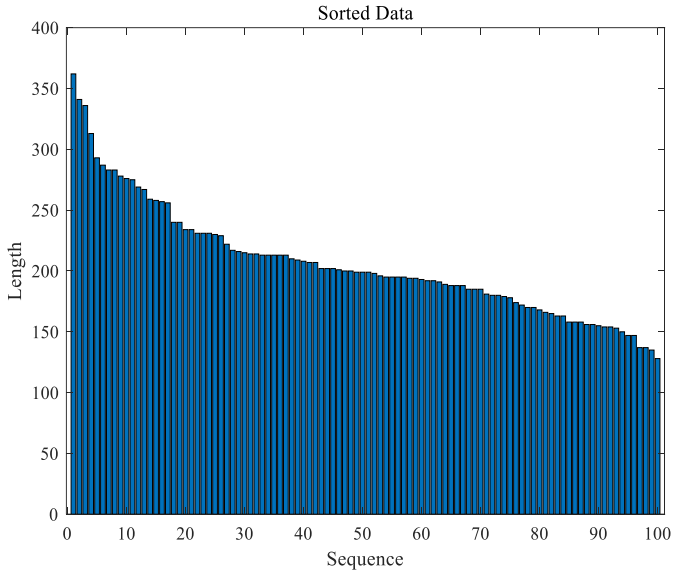

Figure 7. Sorted data.

The pre-processed high-dimensional time series are input into the CNN for feature extraction. The original data has 21 features, and 17 features were obtained after the dimensionality reduction process. Then, the dimensionality reduction features are input into the BiLSTM for temporal information learning. The changes in training error during the training process are shown in **Figure 8**, and the RMSE of training gradually tends to be smooth as the number of training periods increases.

The RUL prediction results for four randomly selected engines in the FD003 dataset are shown in **Figure 8**. The deep convolutional structure of the CNN-BiLSTM model can effectively extract the deep features of engine degradation. However, the initial historical data is small and difficult to predict. However, with the increase of running time and data, the model combines the advantages of spatial and time features, which can effectively improve the prediction accuracy over a long period of time.

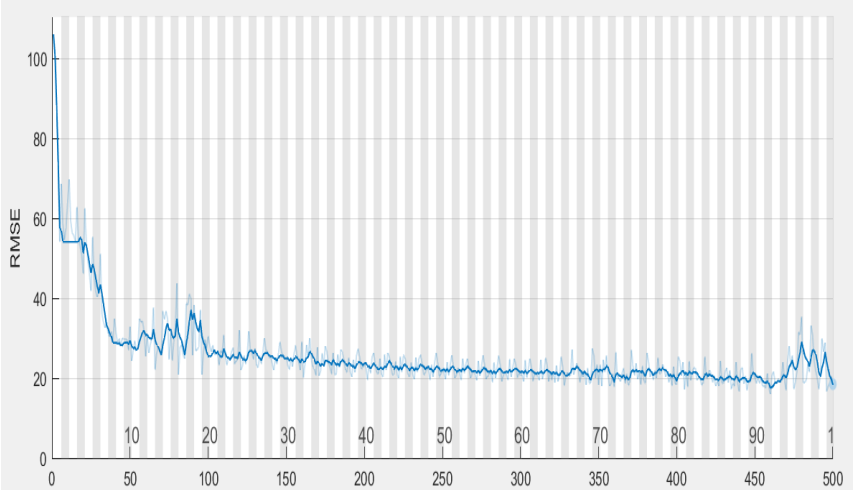

Figure 8. Model training process.

5.4. Model comparison analysis

Step 1: Construction of HI

Firstly, FD002 data from working state six is utilized as the training set to construct the HI model in the source domain, while FD003 data from working state one serves as the training set for constructing the HI model in the target domain. The HI curves in both source and target domains undergo processing with a sliding time window of size 30, which are then inputted into the CNN to extract the engine degradation features and obtain the HI. Subsequently, these results are fed into the BiLSTM based on transfer learning for RUL prediction. The effectiveness of feature extraction using both the CNN method and the principal component analysis (PCA) method is compared by using correlation (Corr) and monotonicity (Mon) metrics. The results are displayed in **Table 3**.

Table 3. Evaluation of two methods of HI construction.

Methods	Corr	Mon
PCA	0.85	0.15
CNN	0.93	0.25

As can be observed from **Table 3**, the values of Corr and Mon achieved by the HI method, as constructed using the CNN prediction model, surpass those of the PCA method. This validates that the method developed in this paper is capable of effectively extracting features, which significantly contributes to enhancing the prediction accuracy of the model.

Step 2: RUL prediction of CNN-BiLSTM based on transfer learning

In this paper, the RUL prediction results of one full test cycle of the No.53 unit, which has more test cycles, are selected. Then, the RMSE metric, which measures the error between the predicted and true values, is calculated according to Equation (31), and the histogram of prediction error based on RMSE is given in **Figure 9**.

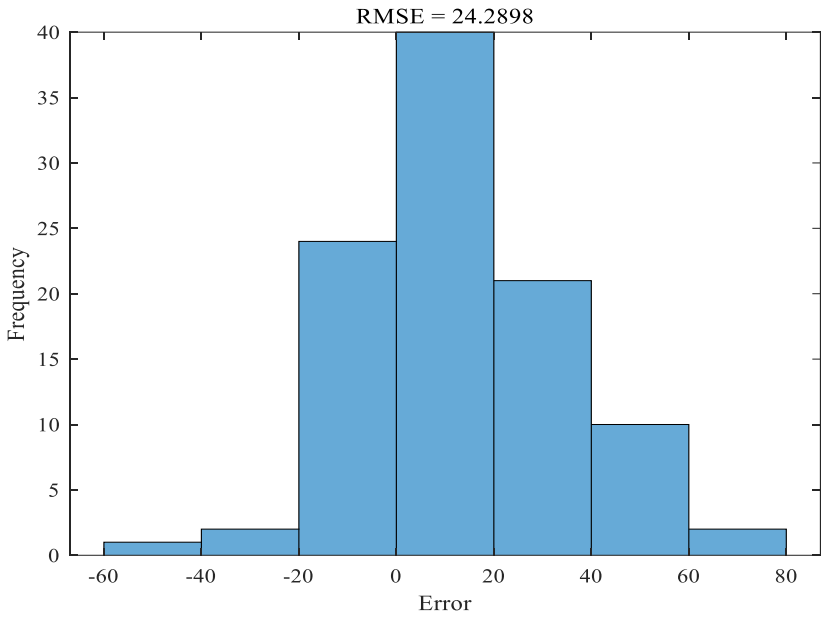

Figure 9. Histogram of prediction error.

Finally, the prediction results have been estimated by the MC simulation technique for the interval. The 95% confidence interval of the RUL prediction obtained by the MC simulation technique for the No.53 engine is given in **Figures 9** and **10**, respectively. The prediction results of RUL for the No.53 engine based on the SVR method are given in **Figure 10**.

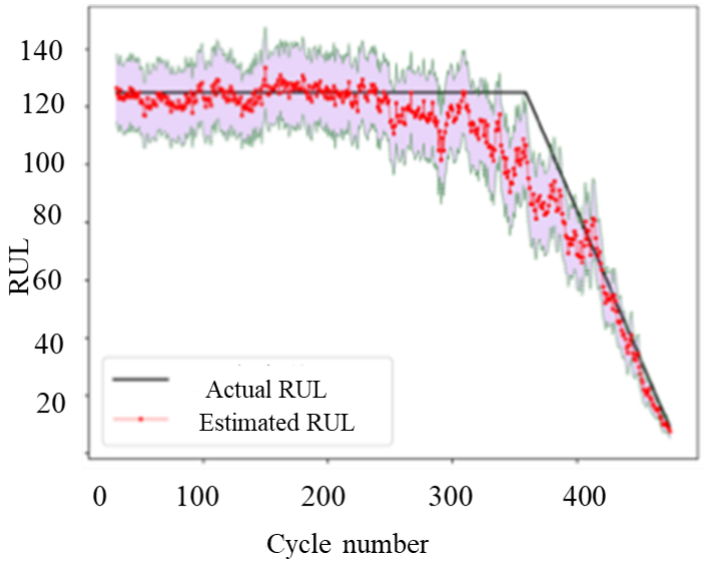

Figure 10. RUL prediction results of Engine 53 based on CNN-BiLSTM.

In order to verify the effectiveness of the proposed method, the CNN model is applied to perform RUL prediction for engine No. 53 of FD003, and the prediction results are shown in **Figure 10**.

By comparing **Figures 11** and **12**, it can be seen that there is a large deviation between the prediction and the true value at the early stage of the operation due to the small amount of performance monitoring data. However, the proposed method in this

paper is able to capture its degradation features better with the help of the source domain data, and therefore, the prediction results are better. The traditional prediction method SVR only provides a single point estimate rather than a probability distribution when predicting the URL, so the uncertainty of the prediction result cannot be measured, which increases the risk of maintenance decisions.

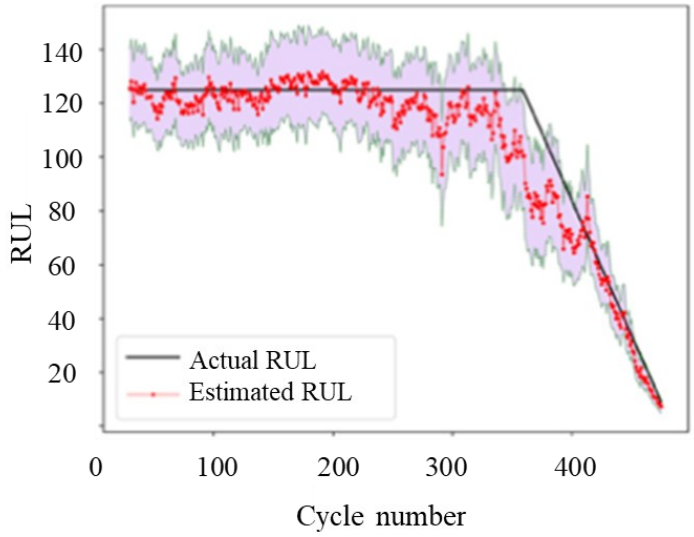

Figure 11. RUL prediction results of engine 53 based on CNN.

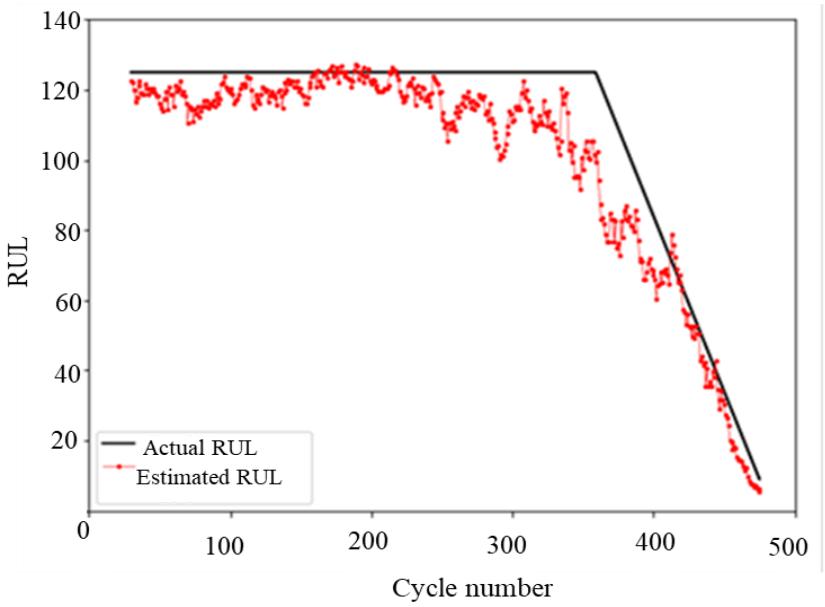

Figure 12. RUL prediction results of engine 53 based on SVR.

Step 3: Uncertainty Metric

Each engine in the dataset has a different degree of initial wear, and a large amount of random noise is introduced into the data, leading to inevitable uncertainty in the RUL prediction results. Therefore, it is necessary to quantify the uncertainty of the RUL prediction process, which in turn can guide the next maintenance decision. The RUL prediction results based on CNN-BiLSTM and LSTM models for the four engines in the test set with different amounts of monitoring data are presented in

Tables 4 and **5**, respectively. Moreover, the 95% confidence intervals (CI) of RUL are calculated by the MC simulation technique.

Table 4. RUL prediction results based on CNN-BiLSTM model for different cycles.

Engine	Cycle number	Actual RUL	Estimated RUL	95% CI
1	31	112	103.3	[85.3, 121.5]
11	83	97	89.2	[61.5, 117.9]
17	165	50	47.6	[39.6, 55.6]
34	203	7	6.5	[3.5, 9.3]

Table 5. RUL prediction results based on the CNN model for different cycles.

Engine	Cycle number	Actual RUL	Estimated RUL	95% CI
1	31	112	98.9	[62.3, 135.5]
11	83	97	85.7	[51.5, 119.7]
17	165	50	45.1	[30.6, 60.3]
34	203	7	5.9	[3.0, 9.5]

From the comparison between **Tables 4** and **5**, it can be seen that at the early stage of engine degradation. There are large deviations between the predicted and true values based on the CNN model and SVR model due to the small number of training samples. The CNN-BiLSTM model proposed in this paper can effectively extract information on the degradation mechanism from the data of different operating conditions, which can improve the accuracy of RUL prediction with the limited amount of training sample data. The training sample size increases with the increase of engine operation cycles, and both prediction methods achieve better prediction results with sufficient training sample data, but the CNN-BiLSTM model proposed in this paper has better prediction results. The interval estimation results in **Table 5**. are valuable for reducing the maintenance decision risk.

To verify the role of temporal characteristics in feature transfer, Task A is selected with seven bearings under operating condition 1 as the source domain and bearings No. 1–3 under operating condition 2 as the target domain. Task B takes 7 bearings under operating condition 1 as the source domain and bearings No. 4–7 under operating condition 2 as the target domain. Three classic domain adaptation methods, SA, GFK, and KMM are used for comparison. As demonstrated in **Figures 13** and **14**, the domain-adaptive feature transfer learning approach surpasses traditional statistical features in both RMSE and MAPE metrics of RUL prediction. This indicates that compared with the conventional method of directly identifying the common feature subspace for sequence data, incorporating the temporal features inherent in the sequence itself aids in enhancing the transfer effect and reducing the RUL prediction error.

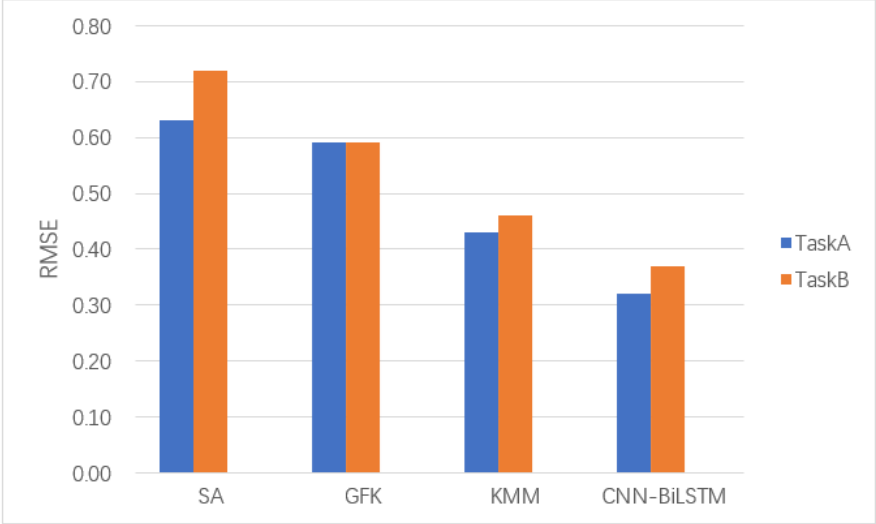

Figure 13. Comparison results of RMSE of RUL for different domain adaptive algorithms.

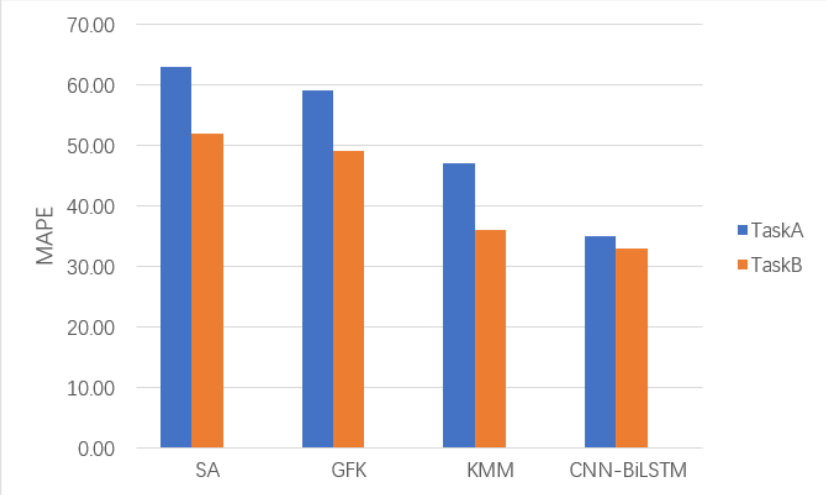

Figure 14. Comparison results of MAPE of RUL for different domain adaptive algorithms.

Finally, four representative RUL prediction methods have been selected for comparison, and the results are shown in **Table 6**. The average RUL prediction error on seven target domain bearings was adopted. The four methods include one shallow model feature selection method and three deep feature learning methods. Cheng et al. [35] have used a CNN to extract features and a BiLSTM to construct a rolling-bearing RUL prediction model. Cheng et al. [36] have used a transferable convolutional neural network to extract time-frequency domain degradation features from bearing vibration signals and achieved RUL prediction. Fu et al. [37] have employed a domain-invariant deep residual LSTM to realize RUL prediction across domains. Yang et al. [38] have proposed an LSTM-based bearing RUL prediction method and improved the prediction accuracy by modifying the dropout module during the training process. Comparing the RMSE error and MAPE error, these two indicators demonstrate the superiority of the proposed methods in the paper.

Table 6. Comparison of RUL prediction results of five different methods.

Methods	Document[32]	Document[33]	Document[34]	Document[35]	CNN-BiLSTM
MAPE	39.51	31.05	35.25	45.15	30.33
RMSE	0.37	0.59	0.33	0.65	0.29

6. Conclusion

In order to solve the problem that insufficient training samples affect the prediction accuracy of the RUL model, this paper presents a CNN-BiLSTM model prediction RUL method based on feature transfer. It is validated using the CMAPSS dataset, leading to the following conclusions:

(1) During the feature extraction stage, On the basis of signal decomposition using local mean values， we initially use CNN to extract degradation features. Subsequently, health factors are constructed by evaluating monotonicity and correlation to refine the features further. Compared to the commonly used PCA dimensionality reduction method, the features extracted using CNN exhibit superior performance in terms of correlation and monotonicity.

(2) In the RUL prediction stage, we have enhanced the conventional LSTM prediction algorithm by employing BiLSTM to model the time series data. This effectively mines the degradation mechanism information from data under different operational conditions. We have introduced the transfer learning algorithm to tackle the problem of different data distributions caused by non-uniform working conditions of mechanical equipment data. Consequently, a transfer learning-based BiLSTM method has been proposed to solve the challenge of RUL prediction under multiple working conditions.

(3) The confidence interval of the RUL is estimated using the Monte Carlo simulation technique. Our proposed method effectively addresses the difficulty of measuring the uncertainty of model prediction results, which holds significant practical value for reducing the risk associated with maintenance decisions.

Although the rolling bearing RUL prediction model constructed in this paper performs well for different bearings, there is a problem of prediction lag for some bearings. Therefore, subsequent research will optimize the GCN-LSTM model structure to improve the prediction effect of rolling bearing RUL.

Author contributions: Conceptualization, ZJ and WY; methodology, ZJ and WY; software, ZJ; formal analysis, ZJ and YZ; data curation, ZJ and YZ; writing original draft preparation, ZJ and YZ; writing review and editing, ZJ and YZ; supervision, WY; project administration, YZ. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the Key R&D and Transformation Plan of Qinghai Province (2022-QY-218), and the National Natural Science Foundation of China (62102262).

Acknowledgments: This paper's logical organization and content quality have been enhanced. Thank anonymous reviewers and journal editors for assistance.

Conflict of interest: The authors declare no conflict of interest.

References

1. Khan A, Azad MM, Sohail M, et al. A Review of Physics-based Models in Prognostics and Health Management of Laminated Composite Structures. International Journal of Precision Engineering and Manufacturing-Green Technology. 2023; 10(6): 1615-1635. doi: 10.1007/s40684-023-00509-4
2. Nguyen KTP, Medjaher K, Tran DT. A review of artificial intelligence methods for engineering prognostics and health management with implementation guidelines. Artificial Intelligence Review. 2022; 56(4): 3659-3709. doi: 10.1007/s10462-022-10260-y
3. Cai B, Fan H, Shao X, et al. Remaining useful life re-prediction methodology based on Wiener process: Subsea Christmas tree system as a case study. Computers & Industrial Engineering. 2021; 151: 106983. doi: 10.1016/j.cie.2020.106983
4. Ouyang T, Wang C, Xu P, et al. Prognostics and health management of lithium-ion batteries based on modeling techniques and Bayesian approaches: A review. Sustainable Energy Technologies and Assessments. 2023; 55: 102915. doi: 10.1016/j.seta.2022.102915
5. Deng W, Nguyen KTP, Medjaher K, et al. Physics-informed machine learning in prognostics and health management: State of the art and challenges. Applied Mathematical Modelling. 2023; 124: 325-352. doi: 10.1016/j.apm.2023.07.011
6. Chen J, Huang R, Chen Z, et al. Transfer learning algorithms for bearing remaining useful life prediction: A comprehensive review from an industrial application perspective. Mechanical Systems and Signal Processing. 2023; 193: 110239. doi: 10.1016/j.ymssp.2023.110239
7. Li X, Yu D, Søren Byg V, et al. The development of machine learning-based remaining useful life prediction for lithium-ion batteries. Journal of Energy Chemistry. 2023; 82: 103-121. doi: 10.1016/j.jechem.2023.03.026
8. Gebraeel N, Lei Y, Li N, et al. Prognostics and remaining useful life prediction of machinery: advances, opportunities and challenges. Journal of Dynamics, Monitoring and Diagnostics. 2023; 1-12.
9. Si X, Li T, Zhang Q, et al. Prognostics for Linear Stochastic Degrading Systems with Survival Measurements. IEEE Transactions on Industrial Electronics. 2020; 67(4): 3202-3215. doi: 10.1109/tie.2019.2908617
10. Zhang J, Li X, Tian J, et al. An integrated multi-head dual sparse self-attention network for remaining useful life prediction. Reliability Engineering & System Safety. 2023; 233: 109096. doi: 10.1016/j.ress.2023.109096
11. Wang B, Lei Y, Li N, et al. A Hybrid Prognostics Approach for Estimating Remaining Useful Life of Rolling Element Bearings. IEEE Transactions on Reliability. 2020; 69(1): 401-412. doi: 10.1109/tr.2018.2882682
12. Li Y, Chen Y, Hu Z, et al. Remaining useful life prediction of aero-engine enabled by fusing knowledge and deep learning models. Reliability Engineering & System Safety. 2023; 229: 108869. doi: 10.1016/j.ress.2022.108869
13. Dong S, Xiao J, Hu X, et al. Deep transfer learning based on Bi-LSTM and attention for remaining useful life prediction of rolling bearing. Reliability Engineering & System Safety. 2023; 230: 108914. doi: 10.1016/j.ress.2022.108914
14. Wang Z, Ta Y, Cai W, et al. Research on a remaining useful life prediction method for degradation angle identification two-stage degradation process. Mechanical Systems and Signal Processing. 2023; 184: 109747. doi: 10.1016/j.ymssp.2022.109747
15. Wei Y, Wu D, Terpenny J. Bearing remaining useful life prediction using self-adaptive graph convolutional networks with self-attention mechanism. Mechanical Systems and Signal Processing. 2023; 188: 110010. doi: 10.1016/j.ymssp.2022.110010
16. Tian, Zhigang, R. Wong, and R. Safaei. A neural network approach for remaining useful life prediction utilizing both failure and suspension histories. Mechanical Systems & Signal Processing 24.5(2010):1542-1555.
17. Guo L, Lei Y, Xing S, et al. Deep Convolutional Transfer Learning Network: A New Method for Intelligent Fault Diagnosis of Machines with Unlabeled Data. IEEE Transactions on Industrial Electronics. 2019; 66(9): 7316-7325. doi: 10.1109/tie.2018.2877090
18. Pei H. Review of Machine Learning Based Remaining Useful Life Prediction Methods for Equipment. Journal of Mechanical Engineering. 2019; 55(8): 1. doi: 10.3901/jme.2019.08.001
19. Li Z, Wu D, Hu C, et al. An ensemble learning-based prognostic approach with degradation-dependent weights for remaining useful life prediction. Reliability Engineering & System Safety. 2019; 184: 110-122. doi: 10.1016/j.ress.2017.12.016

20. Xu D, Xiao X, Liu J, et al. Spatio-temporal degradation modeling and remaining useful life prediction under multiple operating conditions based on attention mechanism and deep learning. Reliability Engineering & System Safety. 2023; 229: 108886. doi: 10.1016/j.ress.2022.108886
21. Lei Y. Deep Transfer Diagnosis Method for Machinery in Big Data Era. Journal of Mechanical Engineering. 2019; 55(7): 1. doi: 10.3901/jme.2019.07.001
22. Yu Y, Hu C, Si X, et al. Averaged Bi-LSTM networks for RUL prognostics with non-life-cycle labeled dataset. Neurocomputing. 2020; 402: 134-147. doi: 10.1016/j.neucom.2020.03.041
23. Ellefsen AL, Ushakov S, Zhang HX. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. Reliability Engineering & System Safety. 2019; 183: 240-251. doi: 10.1016/j.ress.2018.11.027
24. Guo F, Wu X, Liu L, et al. Prediction of remaining useful life and state of health of lithium batteries based on time series feature and Savitzky-Golay filter combined with gated recurrent unit neural network. Energy. 2023; 270: 126880. doi: 10.1016/j.energy.2023.126880
25. Xu Z, Bashir M, Liu Q, et al. A novel health indicator for intelligent prediction of rolling bearing remaining useful life based on unsupervised learning model. Computers & Industrial Engineering. 2023; 176: 108999. doi: 10.1016/j.cie.2023.108999
26. Kamei S, Taghipour S. A comparison study of centralized and decentralized federated learning approaches utilizing the transformer architecture for estimating remaining useful life. Reliability Engineering & System Safety. 2023; 233: 109130. doi: 10.1016/j.ress.2023.109130
27. Mao W, He J, Zuo MJ. Predicting Remaining Useful Life of Rolling Bearings Based on Deep Feature Representation and Transfer Learning. IEEE Transactions on Instrumentation and Measurement. 2020; 69(4): 1594-1608. doi: 10.1109/tim.2019.2917735
28. Chen J, Mao W, Liu J, et al. On line residual life assessment of rolling bearings under unknown working conditions based on recursive prediction of time series transfer. Control and Decision. 2021; 36(2): 3-1.
29. Chen J, Mao W, Liu J, et al. Prediction method of bearing remaining useful life based on depth time series feature transfer. Control and Decision. 2021; 36(7): 8-17.
30. Li J, Chen Y, Xiang H, et al. Prediction of remaining useful life of aeroengines based on LSTM-DBN. Systems Engineering and Electronics Technology. 2020; 42(7): 1637-1644.
31. Hu CH, Pei H, Si XS, et al. A Prognostic Model Based on DBN and Diffusion Process for Degrading Bearing. IEEE Transactions on Industrial Electronics. 2020; 67(10): 8767-8777. doi: 10.1109/tie.2019.2947839
32. Zhang Z, Dong R, Lan G, et al. Diesel particulate filter regeneration mechanism of modern automobile engines and methods of reducing PM emissions: a review. Environmental Science and Pollution Research. 2023; 30(14): 39338-39376. doi: 10.1007/s11356-023-25579-4
33. Li S, Chen H, Chen Y, et al. Hybrid Method with Parallel-Factor Theory, a Support Vector Machine, and Particle Filter Optimization for Intelligent Machinery Failure Identification. Machines. 2023; 11(8): 837. doi: 10.3390/machines11080837
34. Duan W, Song S, Xiao F, et al. Battery SOH estimation and RUL prediction framework based on variable forgetting factor online sequential extreme learning machine and particle filter. Journal of Energy Storage. 2023; 65: 107322. doi: 10.1016/j.est.2023.107322
35. Cheng Y, Hu K, Wu J, et al. A convolutional neural network based degradation indicator construction and health prognosis using bidirectional long short-term memory network for rolling bearings. Advanced Engineering Informatics. 2021; 48: 101247. doi: 10.1016/j.aei.2021.101247
36. Cheng H, Kong X, Chen G, et al. Transferable convolutional neural network based remaining useful life prediction of bearing under multiple failure behaviors. Measurement. 2021; 168: 108286. doi: 10.1016/j.measurement.2020.108286
37. Fu S, Zhang Y, Lin L, et al. Deep residual LSTM with domain-invariance for remaining useful life prediction across domains. Reliability Engineering & System Safety. 2021; 216: 108012. doi: 10.1016/j.ress.2021.108012
38. Yang J, Peng Y, Xie J, et al. Remaining Useful Life Prediction Method for Bearings Based on LSTM with Uncertainty Quantification. Sensors. 2022; 22(12): 4549. doi: 10.3390/s22124549