Precision Mechanics & Digital Fabrication
https://www.acadlore.com/journals/PMDF

A Multi-Scale Temporal Convolutional Network Approach for Remaining Useful Life Prediction of Rolling Bearings

Tichun Wang*, Qiji Teng

College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, 210000 Nanjing, China

* Correspondence: Tichun Wang (wangtichun2010@nuaa.edu.cn)

Received: 01-06-2025 **Revised:** 02-24-2025 **Accepted:** 03-02-2025

Citation: T. C. Wang and Q. J. Teng, "A Multi-Scale Temporal Convolutional Network approach for Remaining Useful Life prediction of rolling bearings," *Precis. Mech. Digit. Fabr.*, vol. 2, no. 1, pp. 31–43, 2025. https://doi.org/10.56578/pmdf020103.

Abstract: Rolling bearings, as key components of rotating machinery, play a crucial role in the reliable operation of equipment. Over time, rolling bearings inevitably experience wear and fatigue, leading to damage. Accurate prediction of their Remaining Useful Life (RUL) is of paramount importance. This paper proposes an RUL prediction model based on the Multi-Scale Temporal Convolutional Network (MSTCN). The model effectively integrates both time-domain and frequency-domain information from bearing vibration signals through a multi-scale feature extraction module, enabling it to capture feature representations at different time scales. Additionally, the MSTCN's powerful temporal modeling capabilities allow it to capture long-term dependencies and short-term fluctuations in the bearing degradation process. Experimental results show that, compared to traditional methods, the proposed MSTCN model significantly improves the accuracy and stability of RUL predictions on the PHM2012 bearing dataset, demonstrating the effectiveness of the method in predicting the RUL of rolling bearings.

Keywords: Rolling bearings; Remaining Useful Life (RUL) prediction; Multi-Scale Temporal Convolutional Network (MSTCN); Intelligent design

1 Introduction

Rolling bearings play a crucial role in the transmission system [1]. They are key components that connect rotating and fixed parts, mainly serving to support the rotating parts, transmit loads, and reduce friction [2]. Therefore, accurately extracting the features of rolling bearings and predicting their RUL is of significant importance for ensuring the safety performance of equipment [3, 4].

RUL prediction methods can be categorized into three types based on their fundamental techniques and methods: physics-based methods [5, 6], data-driven methods [7, 8], and hybrid methods [9, 10]. Physics-based methods typically rely on a deep understanding of the physical characteristics and working principles of the equipment or system [11, 12]. These methods build physical models to describe the wear, failure, and degradation process of the equipment, and predict the RUL based on these models. In contrast, data-driven methods [13, 14] do not depend on the physical models of the equipment but instead model the degradation process of the equipment by analyzing historical data. These methods usually use a large amount of monitoring data to identify degradation patterns and further predict RUL. Hybrid methods combine the advantages of both physics-based and data-driven approaches, usually by integrating physical models with data-driven techniques to improve the accuracy and reliability of RUL predictions.

In recent years, with the rapid development of deep learning and neural networks, deep learning and hybrid methods for rolling bearing RUL prediction have gradually become a research hotspot. Qian [15] proposed an RUL prediction method for rotating components based on nonlinear phase space reconstruction theory in chaotic time series analysis. This method models the degradation process of rotating components using nonlinear phase space reconstruction theory, which can capture the nonlinearity and uncertainty in the degradation process, thereby improving the accuracy and reliability of life prediction. Liu et al. [16] proposed an RUL prediction method based on health indicators, where the method adaptively extracts the feature distances between normal and degraded samples to construct the health indicator of the rolling bearing. Afterward, Liu et al. [16] combined an unsupervised clustering algorithm to determine the alarm threshold and fault threshold, adaptively setting reasonable early warning standards. Based on this, a four-parameter exponential model and particle filtering algorithm were used for real-time tracking and prediction of the RUL of rolling bearings. Yao et al. [17] proposed a fan noise field RUL prediction method combining an acoustic model and neural networks by applying FW-H acoustic analogy theory to calculate the fan's noise field solution. This method trained the sample data using the BP neural network algorithm. The trained neural network could predict the noise field of the axial fan during operation, thus achieving accurate prediction of fan noise and indirectly inferring the RUL of the fan. Ren et al. [18] proposed a novel RUL prediction method based on deep convolutional neural networks (CNN). This method introduced the spectrum-principal-energy-vector (SPEV) as a new feature extraction technique, and by using deep CNNs, these feature vectors were input into the network for training. The CNN model can automatically learn complex patterns in the signal through multiple convolution layers, thus improving the accuracy of RUL prediction. With this deep learning method, Ren et al. [18] successfully enhanced the prediction accuracy of equipment life.

This paper proposes an RUL prediction method for rolling bearings based on MSTCN. This method first extracts key features from the vibration signals through feature processing, and then, through the MSTCN's modeling capability, effectively extracts multi-level information from the features, mining potential temporal features and hidden patterns, thereby predicting the RUL of rolling bearings. The main innovations of this method are as follows:

(1) An adaptive receptive field adjustment mechanism is designed. Different convolution layers of the MSTCN can dynamically adjust the receptive field size according to the characteristic changes in the degradation stages of the rolling bearing. The early stage quickly captures the overall trend of the signal, while the later stage finely explores the local details, effectively improving the model's adaptability and prediction accuracy for different degradation stages.

(2) The feature extraction of the rolling bearings allows for more effective integration of information from different scales, suppressing the interference of redundant features and improving the robustness and accuracy of the prediction model.

(3) The advantages of residual learning are fully utilized to construct a deeper MSTCN network. Residual connections effectively mitigate the gradient vanishing problem during deep network training, enabling the model to learn more complex temporal dependencies, thus improving the accuracy of rolling bearing RUL prediction.

2 Construction of the MSTCN Prediction Method

2.1 Dilated Causal Convolution

Dilated causal convolution is an extension of causal convolution, where a dilation factor is introduced by adding gaps between the elements of the convolution kernel to expand the receptive field. This allows the model to convolve over a larger time range without increasing the size of the convolution kernel or the network depth. Its network structure is shown in Figure 1.

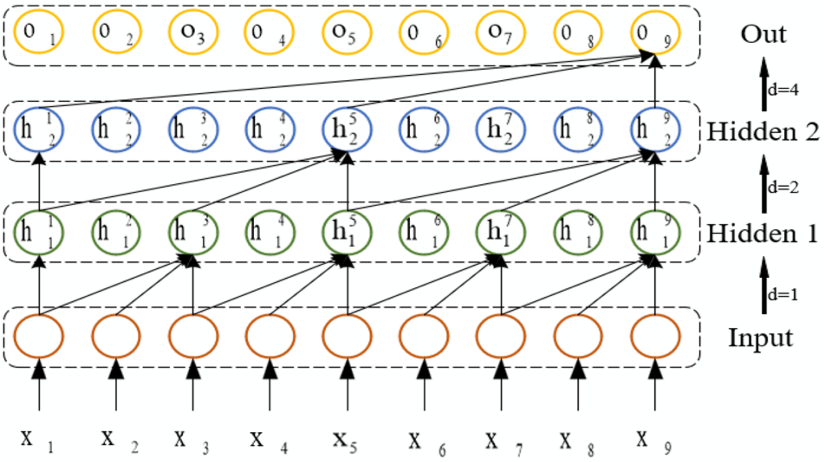

Out
d=4
Hidden 2
d=2
Hidden 1
d=1
Input
o_1 o_2 o_3 o_4 o_5 o_6 o_7 o_8 o_9
h_2^1 h_2^2 h_2^3 h_2^4 h_2^5 h_2^6 h_2^7 h_2^8 h_2^9
h_1^1 h_1^2 h_1^3 h_1^4 h_1^5 h_1^6 h_1^7 h_1^8 h_1^9
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9

Figure 1. Dilated causal convolution network structure

For a one-dimensional input sequence $x = [x_1, x_2, \ldots, x_t]$, the output $F(s)$ of the causal convolution at time step t can be expressed as:

$$F(s) = \sum_{i=0}^{k-1} f(i) \times x_{s-d\times i} \tag{1}$$

The length of the dilated convolution kernel is:

$$l = k + (k-1)(d-1) \tag{2}$$

In formulas (1) and (2), $F(.)$ represents the convolution operation, k represents the size of the convolution kernel (shown as $k = 3$ in the figure above), $f(i)$ is the i-th data in the convolution kernel, $s - d \times i$ represents the direction of historical time, d represents the dilation factor. In the dilated causal convolution block, the dilation factor increases in the form of $d = 2^t$.

In Temporal Convolutional Networks (TCN), the receptive field is defined as the range of the input data that each unit in the feature map corresponds to. In other words, it is the range of the input data that each node can "see." Expanding the receptive field can enhance the model's long-term memory capability, as the model can capture longer-term dependencies and patterns. As shown in Figure 1, dilated causal convolution, by introducing a dilation factor d, can significantly expand the receptive field without increasing the number of convolution layers. For example, in Figure 1, the dilation factors for different layers are d=1, d=2, and d=4. This setup allows the neurons in the output layer to cover a longer historical time step, expanding the receptive field with relatively fewer layers, thus addressing the issue of a small receptive field in the initial layers with fewer layers. This also reduces computational overhead and improves computational efficiency.

For this reason, dilated causal convolution expands the receptive field by setting different dilation factors without increasing the number of layers, effectively reducing the number of intermediate layers in the gradient propagation path, thereby alleviating the vanishing gradient problem to some extent.

In the RUL prediction of rolling bearings, dilated causal convolution can be used to process various sensor data, such as vibration signals, temperature signals, and more. By analyzing these long-time series data, it can accurately predict the RUL of rolling bearings. This technology can identify potential fault signals in the early stages, providing a reliable basis for maintenance decisions and avoiding sudden equipment failures, thus reducing maintenance costs.

In summary, the application of dilated causal convolution in rolling bearing RUL prediction not only enhances the model's ability to capture long-term dependencies but also improves computational efficiency and prediction stability, providing strong support for equipment health management.

2.2 Residual Connection

In the TCN structure, the input and output of the dilated causal convolution are added together through a residual connection. The concept of residual connection was first introduced by He et al. [19], and its structure is shown in Figure 2.

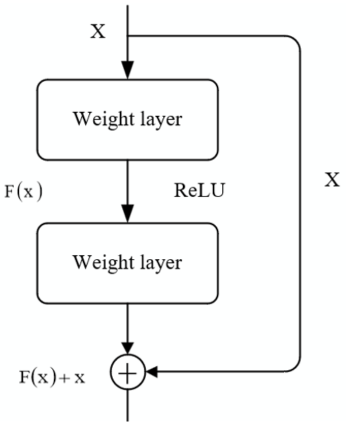


Figure 2. Residual structure diagram

In the convolutional operations of CNN, as the number of layers in the network increases, problems such as vanishing gradients and exploding gradients may occur. In the residual structure, short connections between the input and output are used, which can effectively solve the above problems.

The formula is as follows:

$$y = Activation(x + F(x)) \tag{3}$$

where, x represents the input and $F(\cdot)$ represents the residual function.

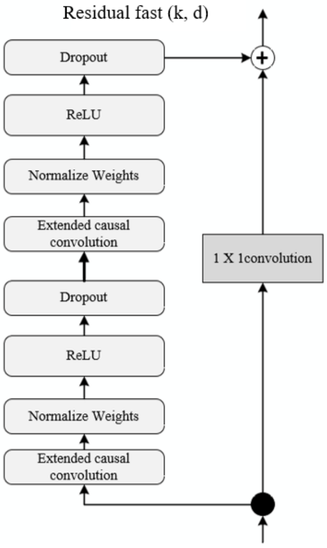


Figure 3. TCN residual structure diagram

However, this structure does not solve the problem of overfitting in the residual structure, so Shao et al. [20] made some improvements to the residual block, and the structure is shown in Figure 3.

From Figure 3, we can see that the TCN residual block consists of two identical submodules and a residual connection. The dilated causal convolution layer is used to capture longer dependencies in the time series. The weight normalization layer standardizes each weight in the network, ensuring that the gradient updates are more stable during training, and preventing issues like gradient explosion or vanishing. Assume there is a weight matrix W, and the formula is:

$$W = g \cdot \frac{v}{\|v\|} \tag{4}$$

where, g is a learnable scalar representing the scale of the weights, v is the direction vector of the weights. The normalized weight is $\frac{v}{\|v\|}$. This way, the network's learning can be independent of the scale of the weights.

The activation function layer introduces nonlinearity so that the neural network can approximate nonlinear relationships. Common activation functions include ReLU, Leaky ReLU, Tanh, Sigmoid, etc. In TCN, the ReLU activation function is used, and the formula is:

$$ReLU(x) = \max(0, x) \tag{5}$$

The Dropout layer is a regularization strategy that randomly "deactivates" part of the neural network neurons during training to alleviate overfitting. In this way, the model does not depend on specific neurons during training, enhancing its generalization ability.

Assume the input is x. In the Dropout layer, each neuron's output will be "dropped" with probability p, meaning it becomes 0, and the remaining part is scaled by a certain factor. The specific operation is as follows:

$$Dropout(x) = \frac{x}{p} \cdot Mask(x) \tag{6}$$

where, $Mask(x)$ is a binary vector with the same dimension as the input, and each element is independently retained with probability p (value is 1) or dropped with probability $1 - p$ (value is 0). During testing, Dropout is usually not applied, or the weights are scaled by a probability of p.

A 1×1 convolutional layer has been added to the main path to adjust the dimensions. This ensures that the input xxx and the output after convolution have the same dimensions, allowing for the addition operation.

The TCN residual block combines dilated causal convolution and short connections, so that TCN can still achieve a large receptive field even with fewer network layers, providing a greater advantage when processing time-series data with long-term historical dependencies.

2.3 Multi-Scale Fusion

MSTCN adopts an innovative structural design consisting of three parallel TCN branches. Each branch uses convolution kernels of different lengths to extract temporal features from the input data, with kernel lengths of 2, 3, and 5, respectively. According to the definition of receptive fields, the receptive field lengths corresponding to these kernels are 5, 9, and 17. This design enables the network to extract features at different time scales from the input data, specifically short-term, medium-term, and long-term temporal features. The short-term features are extracted using smaller receptive fields, capable of capturing rapidly changing temporal information; medium-term features are extracted using slightly larger receptive fields, suitable for capturing smoother changes; and long-term features are extracted using even larger receptive fields to capture long-term trends and periodic changes.

Since the features extracted by each branch contain temporal information at different time scales, simply adding them together may not effectively fuse these features. Therefore, in order to better integrate the multi-scale features extracted by the branches, this study chooses to combine the features from the three branches. This approach fuses the temporal features at different scales along the channel dimension, thus forming a new three-dimensional feature that contains multi-level temporal information. This three-dimensional feature not only contains information from each scale but also provides richer temporal context, which helps the subsequent model processing. The overall structure of MSTCN is shown in Figure 4.

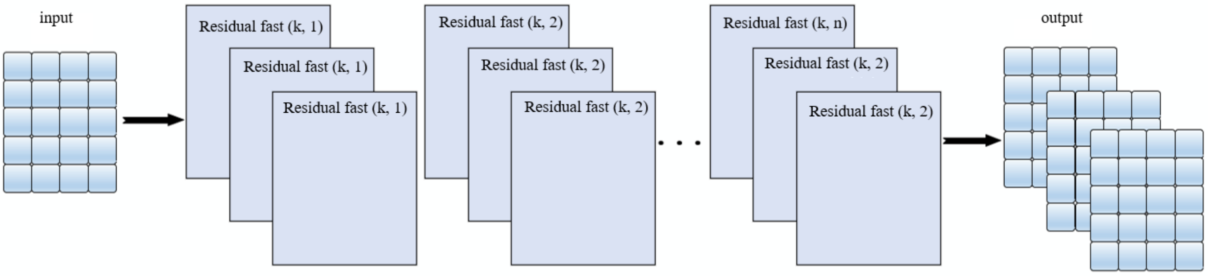


Figure 4. MSTCN structure

The specific computation process is: first, the input data passes through the three parallel TCN branches, each of which uses a convolution kernel of different lengths for feature extraction; then, the extracted features are concatenated along the channel dimension to form a new three-dimensional feature; finally, they are processed and predicted by the subsequent network layers. The formula is as follows:

$$O = concatenate\left(O_1, O_2, O_3\right) \tag{7}$$

where, O_l, O_2, O_3 are the outputs of the three TCN branches, and $O \in R^{C\times S\times T}$ is the final output of MSTCN.

This design enables MSTCN to more comprehensively capture multi-scale information in temporal data and shows better performance when dealing with complex temporal data, thus improving the accuracy of predicting the RUL of rolling bearings.

2.4 MSTCN Prediction Method Structure and Hyperparameter Settings

2.4.1 MSTCN prediction method structure

The structure of the MSTCN prediction method is shown in Figure 5.

As shown in Figure 5, the MSTCN prediction method first extracts time-domain and frequency-domain features from the vibration signal, then the extracted features are used as the feature set input to the MSTCN prediction model, and the prediction results are output.

Time-domain features include mean, standard deviation, root mean square value, root mean square value, peak value, skewness, kurtosis, peak factor, margin factor, waveform factor, and impulse factor. Frequency-domain features include mean, standard deviation, central value, root mean square value, frequency domain kurtosis, average frequency, stability factor, and variation coefficient.

The multi-layer convolution structure in MSTCN can simultaneously capture short-term and long-term features through different-sized convolution kernels. Small convolution kernels can identify rapidly changing signal patterns, while large convolution kernels can extract slower-changing trends to improve the model's prediction accuracy.

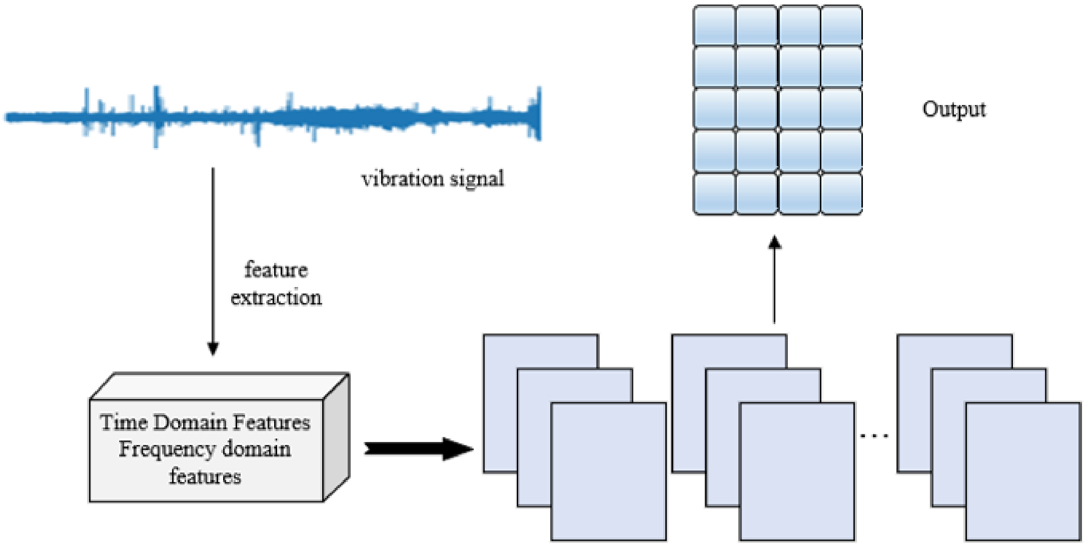


Figure 5. MSTCN method structure

Table 1. Hyperparameter settings

Parameter Name	**Specific Parameter Value**
Optimizer	Adam
Loss Function	Mean Squared Error (MSE)
Dilution Factor (d)	Mean Absolute Error (MAE)
Number of Convolutions	1/2/4
Convolution Kernel Size (k)	32
Learning Rate	3×3
Batch Size	0.001
Epochs	512

2.4.2 MSTCN hyperparameter settings

Hyperparameters play a key role in the performance of the MSTCN model. Different hyperparameter configurations not only affect the training speed of the model but also directly influence its generalization ability. For bearing RUL prediction, appropriate hyperparameter settings can improve the model's ability to capture complex temporal dependencies, thus improving prediction accuracy.

As shown in Table 1, the hyperparameter settings have been experimentally verified to achieve optimal prediction performance on the specific dataset used in this study.

The optimizer in the model is Adam, which effectively handles the training of large datasets, and the global learning rate is set to 0.001. Adam is a very popular and commonly used optimizer because its adaptive learning rate adjustment mechanism allows it to perform well in most tasks. The combination of MSE and MAE loss functions is used to comprehensively measure the model's performance. By setting the dilation factors to 1/2/4, the model can capture features at different time scales, allowing it to capture multi-scale temporal information. The 32 convolution kernels and 3×3 convolution kernel size ensure effective feature extraction. The learning rate is set to 0.001 to allow the model to converge more stably, reducing fluctuations and ensuring stable model updates.

To enhance the model's nonlinear expression ability, a training batch size of 512 and 200 iterations are set to ensure sufficient training of the model. These hyperparameter choices provide a solid foundation for the model's accuracy, stability, and generalization ability.

The combination of the Adam optimizer, small learning rate, MSE and MAE loss functions is suitable for complex time-series prediction tasks. The dilation factors, multi-scale convolution kernels, larger batch size, and sufficient iteration counts all provide the model with enough expressive ability and training space.

3 Experiment

3.1 Experimental Data

The experimental data selected for this study is from the PHM 2012 Data Challenge, which provides the bearing run-to-failure dataset.

The data was collected from the PRO-NOSTIA accelerated degradation platform [21]. This platform can accelerate the degradation of rolling bearings under constant or variable loads, allowing the collection of full-life data of rolling bearings in just a few hours, significantly reducing the data volume. The experimental setup consists of three main components: the rotating mechanism, the load module, and the measurement system. During the experiment, an AC motor drives the rolling bearings to rotate at different speeds, applying different loads for testing. The PRO-NOSTIA accelerated degradation platform is shown in Figure 6.

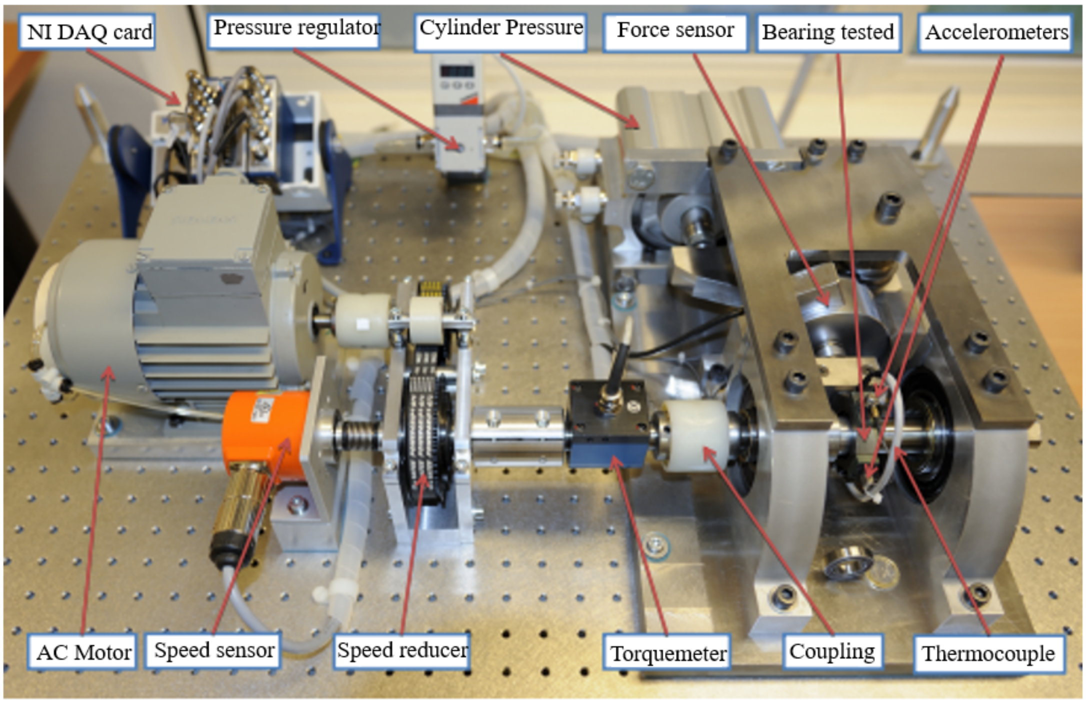


Figure 6. PRO-NOSTIA accelerated degradation platform

During the experiment, two accelerometers were installed in the horizontal and vertical directions of the rolling bearing to collect vibration acceleration data in real time. At the beginning of each experiment, the bearing is in a normal state, and the motor speed and load are increased to accelerate the degradation of the rolling bearing until complete failure.

The accelerometers placed horizontally and vertically on the test bench collect vibration signals from the two directions, namely the x and y directions. The data sampling rate is 25.6 kHz, recording data every 0.1 seconds for 10 seconds, totaling 2560 sample points. For safety, data collection stops when the vibration amplitude exceeds 20 g (20 times the force of gravity, 1 g $\approx$ 9.8 m/s^2). The experiment was conducted under three operating conditions, as shown in Table 2. The detailed information of the bearings under these conditions is shown in Table 2.

Table 2. Bearing data overview

Operating Condition	Load (N)	Speed (rpm)	Training Set	Testing Set
Condition 1	4000	1800	Bearing1_1-Bearing1_6	Bearing1_7
Condition 2	4200	1650	Bearing2_1-Bearing2_6	Bearing2_7
Condition 3	5000	1500	Bearing3_1-Bearing3_2	Bearing3_3

3.2 Experimental Process

In bearing RUL prediction, the RUL prediction problem is treated as a regression problem. During the modeling process, the bearing's operating time is converted into input features for the model, while the RUL percentage is used as the label for RUL prediction. This label design provides a more intuitive reflection of the bearing's degradation throughout its life cycle, thus helping to improve the prediction model's performance. The formula is as follows:

$$y_i = 1 - \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{8}$$

where, x_i is the current operating time of the bearing; $x_{\min}, x_{\max}$ are the initial operating time and the maximum operating time of the bearing, respectively. y_i indicates the degree of degradation, with smaller values of y_i indicating more severe degradation. When $y_i = 0$, the bearing has completely failed.

For the evaluation of the experimental results in this study, in addition to the commonly used MAE and RMSE, a specialized score function, called "score", specifically improved for the PHM 2012 dataset, was also used to evaluate the performance of the developed methods. The formula is as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |er_i| \tag{9}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (er_i)^2} \tag{10}$$

$$score = \frac{1}{n}\sum_{i=1}^{n} S_i \tag{11}$$

where,

$$S_i = \begin{cases} \exp\left(-lin(0.5)\cdot(E_i/5)\right), E_i \leq 0 \\ \exp\left(lin(0.5)\cdot(E_i/20)\right), E_i > 0 \end{cases} \tag{12}$$

$$E_i = \frac{y_i - \hat{y}_i}{y_i} \times 100 \tag{13}$$

In practical engineering operations, overestimation and underestimation of RUL predictions can have different consequences. Underestimation may lead to premature maintenance and unnecessary downtime, which increases costs but carries relatively low risk. On the other hand, overestimation is more dangerous because it may lead to continued operation of equipment without timely maintenance, increasing the risk of failure or even safety accidents [22].

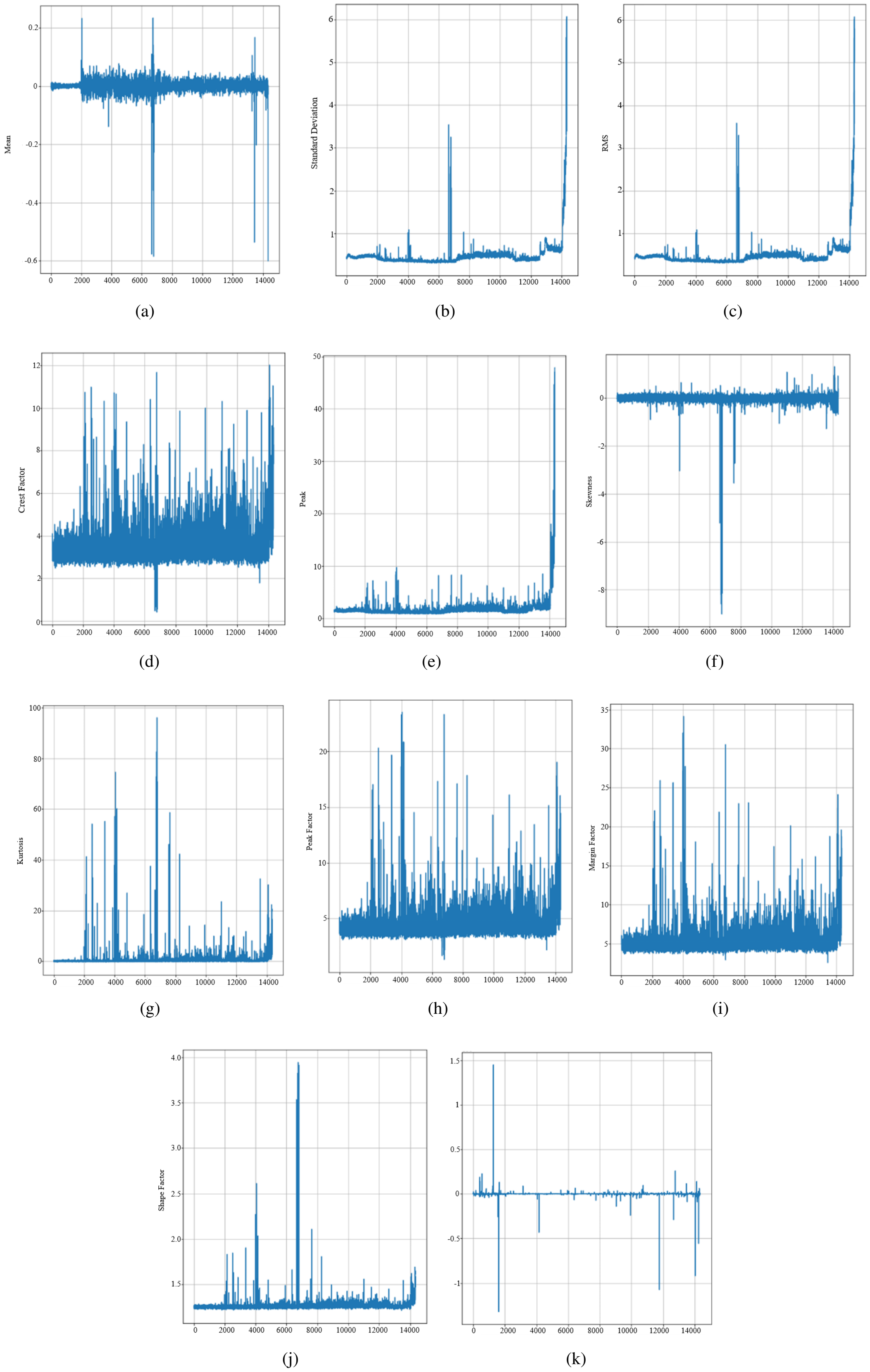
The scoring standard of *Score* assigns higher penalties for overestimation errors to reflect the severity of overestimation, ensuring that the model predicts the equipment's life more cautiously. In the early stages of the equipment's life cycle, the impact of prediction errors is relatively small because the equipment is far from failure, and the risks of inaccurate predictions are lower.

The value of *Score* ranges from 0 to 1. The higher the value of the score, the better the prediction performance. However, in performance evaluation, it is inevitable that the three scoring standards may produce inconsistent results, because MAE and RMSE do not consider the sign of the error (i.e., overestimation or underestimation) and do not differentiate the impact of different errors. On the other hand, the *Score* introduces different penalty mechanisms, assigning different weights to overestimation and underestimation, with particular emphasis on penalizing overestimation errors. This design is more aligned with practical application needs. Therefore, in model selection or evaluation, *Score* is a more appropriate metric.

The x-axis and y-axis signals are stacked to form a new array with dimensions (2660, 2), which is used as input to the model.

The data is first divided into training and testing sets, and then preprocessed before being input into the MSTCN prediction model for computation, and finally, the prediction results are obtained. The hyperparameter settings of the model are listed in Table 1.

The preprocessing process mainly involves normalizing the vibration data and RUL values. The original RUL values can have a wide range and large differences, making it difficult for the model to handle them effectively during training. Normalization compresses the data into the range of (0,1), allowing for more stable weight updates and faster convergence, thus improving the training efficiency of the model. The formula for normalizing the RUL is shown in Eq. (10).



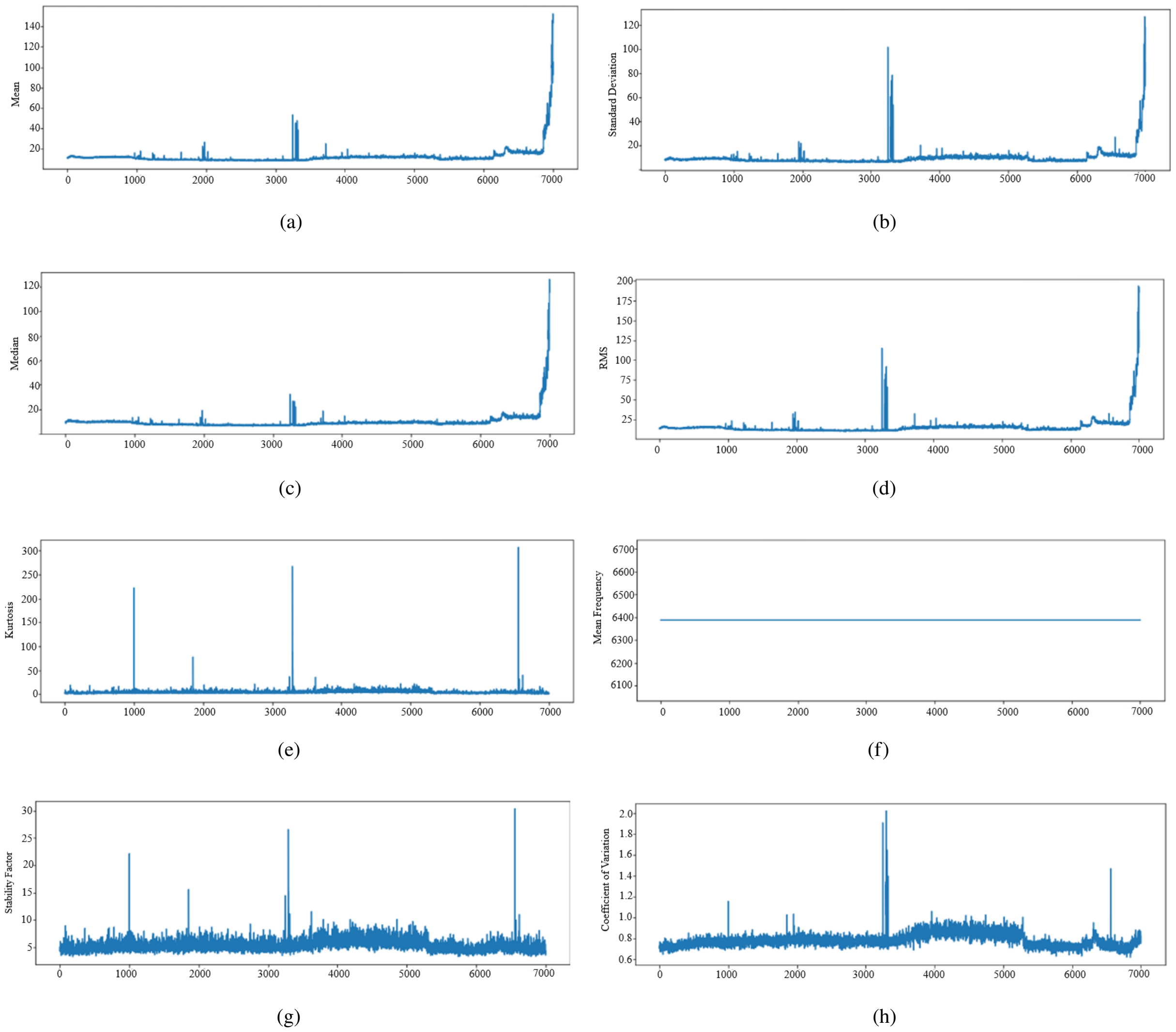
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure 7. Time-domain features: (a) Mean; (b) Standard deviation; (c) Root mean square amplitude; (d) Root mean square value; (e) Peak value; (f) Skewness; (g) Kurtosis; (h) Peak factor; (i) Margin factor; (j) Waveform factor; (k) Impulse factor



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 8. Frequency-domain features: (a) Mean; (b) Standard deviation; (c) Central value; (d) Root mean square value; (e) Frequency-domain kurtosis; (f) Average frequency; (g) Stability factor; (h) Coefficient of variation

In this experiment, time-domain and frequency-domain features of the vibration signals were first extracted. The time-domain features are shown in Figure 7.

Frequency-domain features are shown in Figure 8.

The extracted time-domain and frequency-domain features from Figure 7 and Figure 8 are input into the MSTCN prediction model, and compared with CNN, LSTM, and TCN. The output prediction results are shown in Figure 9.

Figure 9 shows the data from the rolling bearings being input into the four algorithm models for RUL prediction. The black line represents the true RUL curve of the bearing, while the red line represents the predicted RUL curve. The vertical axis shows the normalized RUL.

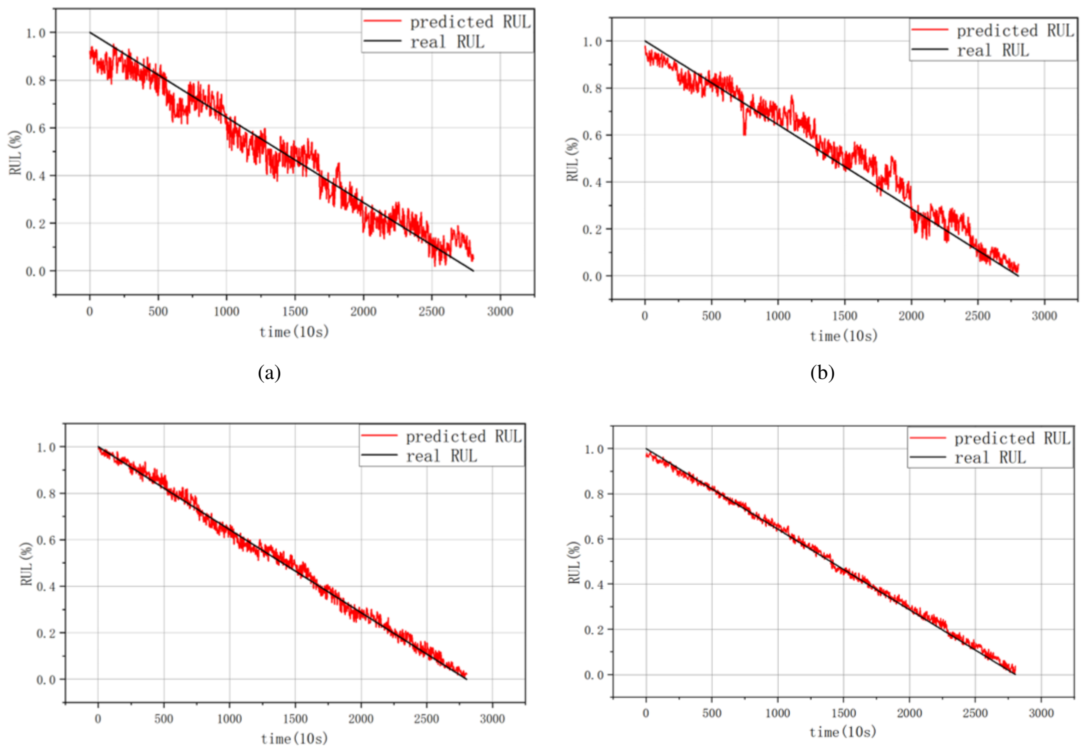
From the figure, it can be observed that CNN shows a certain fluctuation in the overall trend during the RUL prediction of the rolling bearing. Especially in the later stages, the prediction results have a large deviation and fail to accurately reflect the true change in the RUL of the bearing. The LSTM model's prediction performs smoothly in the early stages, capturing the long-term trend of the data well, but shows some deviation later, with the predicted values not accurately following the fluctuations of the true values. The TCN model can more accurately reflect the trend of the RUL change of the rolling bearing during the prediction process, but still exhibits some fluctuation, and there remains a small gap between the predicted and true values. The MSTCN model has the smallest error between the predicted and true values, especially in terms of the trend of the RUL changes, where MSTCN can accurately reflect the true situation. During the prediction process, MSTCN can smoothly track the changes in the RUL of the rolling bearing, avoiding large fluctuations or errors that may occur in other models, demonstrating very high accuracy.

Overall, MSTCN shows the strongest performance in the RUL prediction task and is suitable for the RUL prediction of rolling bearings.

The scores of the four prediction models are given in Table 3.

Table 3 shows the evaluation results of four models (CNN, LSTM, TCN, MSTCN) in the rolling bearing RUL prediction task. The evaluation metrics include MAE, RMSE, and score, with each value reflecting the performance of the models in predicting the RUL of rolling bearings.

From the results in Table 3, it is clear that MSTCN outperforms all other models across all metrics. It has the lowest MAE (0.34) and RMSE (0.41), and the highest score (0.93). This indicates that MSTCN has the best predictive performance for RUL prediction tasks in rolling bearings. TCN follows closely behind, showing good long-term dependency modeling and strong prediction accuracy. LSTM and CNN perform relatively poorly, especially when handling complex vibration signals, showing larger errors and lower predictive accuracy. Overall, MSTCN exhibits significant advantages in RUL prediction due to its multi-scale feature extraction and powerful temporal modeling capabilities.



(a)

(b)

(c)

(d)

Figure 9. Prediction results of four models: (a) Prediction results of the CNN model; (b) Prediction results of the LSTM model; (c) Prediction results of the TCN model; (d) Prediction results of the MSTCN model

Table 3. Score of four prediction models

Prediction Model	MAE	RMSE	Score
CNN	0.49	0.60	0.82
LSTM	0.56	0.51	0.86
TCN	0.38	0.45	0.89
MSTCN	0.34	0.41	0.93

From a combination of Figure 9 and Table 3, it can be concluded that the four models (CNN, LSTM, TCN, MSTCN) show obvious performance differences in rolling bearing RUL prediction. CNN has advantages in local feature extraction but fails to capture long-term dependencies effectively, resulting in large prediction errors and the worst performance. LSTM can capture long-term dependencies in time series data well, but still shows some errors when dealing with complex signals, resulting in moderate performance. TCN models long-term dependencies effectively by stacking convolutional layers, demonstrating higher prediction accuracy and stability, outperforming CNN and LSTM. MSTCN, with its multi-scale feature extraction capability, can capture both short-term and long-term features simultaneously, achieving the best prediction accuracy with the smallest error and highest accuracy. Therefore, MSTCN exhibits the strongest performance in rolling bearing RUL prediction.

4 Conclusions

To address the issues of insufficient model expressive power and difficulty in feature extraction in bearing RUL prediction, this chapter proposes an MSTCN-based method for rolling bearing RUL prediction. The chapter first introduces the construction principle of the MSTCN model, followed by a detailed description of the method's process. By extracting key time-domain and frequency-domain features from vibration signals, MSTCN's powerful modeling capability is employed to capture deep temporal features from the signals, improving the model's prediction accuracy. Finally, through experimental validation, the method effectively enhances the prediction accuracy of RUL in rolling bearings with a single failure mode.

Funding

This research was funded by the National Key Laboratory of Helicopter Aeromechanics Foundation, China (Grant No.: 2023-HA–LB-067-07); the Jiangsu Provincial Natural Science Foundation General Project, China (Grant No.: BK20221481).

Data Availability

The data used to support the research findings are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] M. S. Rathore and S. P. Harsha, "Intelligent fault detection scheme for rolling bearing based on generative adversarial network and auto encoders using convolutional neural network," in *Vibration Engineering and Technology of Machinery*, 2024, pp. 133–153. https://doi.org/10.1007/978-981-99-8986-7_9

[2] D. T. Hoang and H. J. Kang, "A survey on deep learning based bearing fault diagnosis," *Neurocomput.*, vol. 335, pp. 327–335, 2019. https://doi.org/10.1016/j.neucom.2018.06.078

[3] JDMD Editorial Office, N. Gebraeel, Y. Lei, N. Li, X. Si, and E. Zio, "Prognostics and remaining useful life prediction of machinery: Advances, opportunities and challenges," *J. Dyn. Monit. Diagn.*, vol. 2, no. 1, pp. 1–12, 2023. https://doi.org/10.37965/jdmd.2023.148

[4] A. Kumar, C. Parkash, H. S. Tang, and J. W. Xiang, "Intelligent framework for degradation monitoring, defect identification and estimation of remaining useful life (RUL) of bearing," *Adv. Eng. Inform.*, vol. 58, p. 102206, 2023. https://doi.org/10.1016/j.aei.2023.102206

[5] B. Jing, Z. B. Cui, H. D. Sun, X. X. Jiao, and Y. Zhang, "Online life prediction of fuel pumps based on the fusion of failed physics and data-driven methods," *Chin. J. Sci. Instrum.*, vol. 43, no. 3, pp. 68–76, 2022.

[6] X. Y. Chen, H. Q. Zhang, K. Huang, and C. Su, "A review of research on engineering equipment fault prediction methods for predictive maintenance," *Intell. Manuf.*, vol. 2022, no. 2, pp. 50–55, 2022. https://doi.org/10.3969/j.issn.1671-8186.2022.02.011

[7] M. Pecht and J. Gu, "Physics-of-failure-based prognostics for electronic products," *Trans. Inst. Meas. Control*, vol. 31, no. 3–4, pp. 309–322, 2009. https://doi.org/10.1177/0142331208092031

[8] Y. G. Hu, H. Li, P. P. Shi, Z. S. Chai, K. Wang, X. J. Xie, and Z. Chen, "A prediction method for the real-time remaining useful life of wind turbine bearings based on the Wiener process," *Renew. Energy*, vol. 127, pp. 452–460, 2018.

[9] C. Ferreira and G. Gonçalves, "Remaining useful life prediction and challenges: A literature review on the use of machine learning methods," *J. Manuf. Syst.*, vol. 63, pp. 550–562, 2022. https://doi.org/10.1016/j.jmsy.2022.05.010

[10] O. Das, B. D. Duygu, and D. Birant, "Machine learning for fault analysis in rotating machinery: A comprehensive review," *Heliyon*, vol. 9, no. 6, p. e17584, 2023. https://doi.org/10.1016/j.heliyon.2023.e17584

[11] H. M. Xu, Q. Y. Xia, Y. Li, and L. Z. Zhang, "Prediction of remaining life of bearings based on depthwise separable convolutional neural network," *Mech. Strength*, vol. 44, no. 4, pp. 763–771, 2022. https://doi.org/10.16579/j.issn.1001.9669

[12] T. M. Li, X. S. Si, X. Liu, and H. Pei, "Data-model interactive remaining useful life prediction technologies for stochastic degrading devices with big data," *Acta Autom. Sin.*, vol. 48, no. 9, pp. 2119–2141, 2022. https://doi.org/10.16383/j.aas.c201068

[13] X. S. Si, W. Wang, C. H. Hu, and D. H. Zhou, “Remaining useful life estimation – A review on the statistical data driven approaches,” *Eur. J. Oper. Res.*, vol. 213, no. 1, pp. 1–14, 2011. https://doi.org/10.1016/j.ejor.2010.11.018

[14] N. Han, “Prediction of remaining life of two stage rolling bearings based on hybrid filtering,” Ph.D. dissertation, Xi’an University of Technology, Xi’an, China, 2023.

[15] Y. N. Qian, “Research on degradation tracking and fault prediction methods for rotating components in mechanical systems,” Ph.D. dissertation, Southeast University, Nanjing, China, 2015.

[16] X. Y. Liu, G. Chen, Z. J. Cheng, X. K. Wei, and H. Wang, “Convolution neural network based particle filtering for remaining useful life prediction of rolling bearing,” *Adv. Mech. Eng.*, vol. 14, no. 6, 2022. https://doi.org/10.1177/16878132221100631

[17] J. Y. Yao, H. Y. Meng, J. Yang, B. Liang, and J. C. Cheng, “Prediction of axial flow fan aerodynamic noise based on BP artificial neural network,” *J. Nanjing Univ. Nat. Sci.*, vol. 56, no. 6, pp. 900–908, 2020. https://doi.org/10.13232/j.cnki.jnju.2020.06.013

[18] L. Ren, Y. Q. Sun, H. Wang, and L. Zhang, “Prediction of bearing remaining useful life with deep convolution neural network,” *IEEE Access*, vol. 6, pp. 13 041–13 049, 2018. https://doi.org/10.1109/ACCESS.2018.2804930

[19] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90

[20] J. B. Shao, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv preprint*, 2018. https://doi.org/10.48550/arXiv.1803.01271

[21] P. Nectoux, R. Gouriveau, K. Medjaher, E. Ramasso, B. Chebel-Morello, N. Zerhouni, and C. Varnier, “PRONOSTIA: An experimental platform for bearings accelerated degradation tests,” in *IEEE International Conference on Prognostics and Health Management (PHM’12)*, Denver, Colorado, United States, 2012, pp. 1–8.

[22] T. C. Wang, Q. J. Teng, and G. H. Jin, “A remaining useful life prediction method for rolling bearings based on broad learning system - Multi-scale temporal convolutional network,” *Precis. Mech. Digit. Fabr.*, vol. 1, no. 3, pp. 145–157, 2024. https://doi.org/10.56578/pmdf010303