Remaining Useful Life Prediction in Smart Manufacturing Systems Using a CNN-BiLSTM Model with Attention Mechanism

Nanmei Zhang
School of Intelligent Manufacturing, Anhui Wenda University of Information Engineering, Hefei 231201, China
E-mail: znm19931220@163.com

Keywords: intelligent manufacturing, AI-driven, automation, predictive maintenance

Received: March 14, 2025

With the continuous development of intelligent manufacturing, the maintenance strategy of equipment is also constantly improving, and it is changing from passive maintenance to preventive maintenance and predictive maintenance. Passive maintenance is to perform repairs after equipment fails or shuts down, and this method requires a long downtime maintenance time, resulting in increased maintenance costs. Therefore, this paper combines CNN and BiLSTM to propose an equipment life prediction model, so as to carry out predictive maintenance of equipment through intelligent automation model and improve the prediction accuracy and generalization of intelligent factory equipment RUL. By combining the efficient feature extraction capability of CNN with the sequence data processing advantages of BiLSTM and the weighted redistribution of attention mechanism, the model exhibits excellent performance on multiple data sets. According to the experimental results, it can be seen the advantages of the AM-CNN BiLSTM model are mainly reflected in its high accuracy and stability. On the CWRU dataset, the RMSE value of this model is as low as 0.052, which is better than traditional models, and the prediction accuracy is improved by about 47%. On the UCI dataset, its SCORE value reaches 0.963, indicating stronger generalization ability. All in all, by combining the spatial feature extraction of CNN with the temporal modeling of BiLSTM, and introducing attention mechanism, this model maintains stable performance (fluctuation amplitude<5%) in multi condition data, making it particularly suitable for the analysis and prediction of complex temporal data.

Povzetek: Predstavljen je AM-CNN-BiLSTM za napoved preostale življenjske dobe opreme. Združuje CNN, BiLSTM in pozornost, deluje v cloud-edge okolju, izboljša RMSE in SCORE na CWRU, UCI, Augury, FEMTO ter zagotovi robustno, razložljivo prediktivno vzdrževanje.

1 Introduction

With the development of industrial Internet platform (hereinafter referred to as “platform”) technology, it has become a trend to use industrial Internet of Things technology IoT (Internet of Things) to solve equipment health management problems. On the one hand, it uses the industrial Internet platform OPC UA (OLE for Process Control Unified Architecture) and the management shell AAS (Asset Administration Shell) and other technologies to uniformly encapsulate and transform industrial field equipment protocols [1], establish standard equipment connection and semantic transformation models, and realize efficient connection of massive multi-source heterogeneous equipment, thus improving the efficiency of industrial data collection and processing. On the other hand, the characteristics of big data storage and calculation of industrial Internet platform are used to store and analyze equipment design, manufacturing, and operation data, realize real-time monitoring and early warning analysis of key components of equipment, find faults in advance, and reduce enterprise maintenance costs. At the same time, the platform open sharing technology is used to establish an interoperable interface model to realize information sharing among different equipment manufacturers, thus improving the equipment management level [2].

Traditional equipment health assurance management in the industry mainly focuses on the current technical health status of equipment, and it is mainly based on the models of “post-maintenance” and “planned maintenance”. With the development of equipment health management level, the requirements for real-time, intelligent and prediction ability of current equipment are getting higher and higher [3]. Traditional fault diagnosis methods based on expert knowledge and signal processing are very effective as initial troubleshooting. However, the disadvantage is that there is no early warning in the later stage of the fault, and the whole machine is shut down for maintenance due to untimely replacement of the equipment, which brings huge losses to the enterprise. The core feature of the Industrial Internet is to use edge computing and cloud computing for real-time data analysis and scheduling, and fault diagnosis based on cloud-edge collaboration can reduce fault costs and increase response speed. Through the integration of big data and artificial intelligence and other means, it provides a new enabling platform for online diagnosis and prediction of equipment, so as to predict the fault of equipment health management [4]. On the one hand, massive equipment operating condition data is collected on the edge side. On the other hand, a fault diagnosis and prediction model are established on the platform side for high concurrency model training, and the model is sent to the edge side for real-time diagnosis and prediction, thus forming an effective data and model collaboration and adaptation mechanism and realizing data-driven real-time and comprehensive prediction of equipment and its key components [5, 6].

The industrial internet platform achieves efficient device connection and standardized data application through technologies such as OPC UA and AAS, but there are still some problems in the scenario of device life prediction The sampling frequency and accuracy differences of multi-source devices result in a large amount of noise and missing values in the collected data, and semantic transformation models are difficult to completely eliminate the inconsistency of vendor defined thresholds, which affects the reliability of prediction inputs. The prediction models trained on specific devices experience a significant increase in false positive rates during cross vendor or cross model migration due to differences in degradation mechanisms, requiring frequent re labeling of data and fine-tuning of models, which increases deployment costs. Massive device data needs rapid response from the edge layer, but the heterogeneity of industrial field protocols aggravates the data processing delay. When edge computing resources are limited, it is difficult to meet the timeliness requirements of life prediction The CNN BiLSTM model effectively compensates for the shortcomings of the platform in life prediction by integrating spatial feature extraction and temporal dependency modeling.

The equipment intelligent prediction model can predict the upcoming equipment failure in real time, and provide the relevant information of equipment parts that need to be replaced in time before the equipment failure may occur, so as to effectively reduce the equipment failure rate and effectively save the equipment support management cost, reduce the enterprise equipment operation and maintenance cost, and realize the change of enterprise mode from planned repair to preventive maintenance.

Combining CNN and BiLSTM to construct a device lifespan prediction model can leverage their complementary advantages. CNN excels at extracting local spatiotemporal features from raw sensor data (such as vibration and temperature signals) and capturing short-term abnormal patterns during device degradation. BiLSTM models long-term temporal dependencies through a bidirectional gating mechanism, which can trace historical degradation trends (such as slow wear) and correlate potential future fault symptoms. This combination solves the limitations of a single model - pure CNN is difficult to model long-term degradation laws, and pure RNN models have insufficient feature abstraction ability for the original signal. Therefore, end-to-end optimization is achieved in the two key links of feature extraction and time series prediction, significantly improving prediction accuracy and robustness.

This paper combines CNN and BiLSTM to propose an equipment life prediction model, so as to carry out predictive maintenance of equipment through intelligent automation model and improve the prediction accuracy and generalization of intelligent factory equipment RUL. By combining the efficient feature extraction capability of CNN with the sequence data processing advantages of BiLSTM and the weighted redistribution of attention mechanism, the model exhibits excellent performance on multiple data sets. According to the experimental results, it can be seen that the constructed regression prediction model is superior to other methods in terms of RMSE index. Among them, the prediction accuracy of combined training is higher than that of grouping training, which improves the prediction accuracy.

2 Related works

In the equipment fault warning model, discussing predictive maintenance (PdM) first and then troubleshooting is essentially following the industrial maintenance logic loop of "monitoring → diagnosis → disposal". Predictive maintenance identifies equipment anomalies in advance through real-time data analysis and AI algorithms, providing precise targeted targets for troubleshooting. Moreover, troubleshooting is based on the health indicators and fault characteristics output by PdM, implementing standardized maintenance processes. This sequential design not only avoids the resource waste of "blind maintenance", but also continuously optimizes the model accuracy through the "prediction disposal feedback" loop, forming a closed-loop management from data perception to problem solving.

(1) Predictive maintenance

The basic principle of predictive maintenance technology is to monitor the status of industrial equipment in real time through various sensors, predict possible failures of equipment, and provide accurate modification suggestions for maintainers. Because of its predictability and accuracy, it has attracted the research enthusiasm of many experts, scholars and companies and factories.

Data-driven approaches and experience-based approaches are similar in some ways. However, the data-driven method does not need prior knowledge and does not pay attention to the internal situation of the prediction model. Compared with other methods, it is simpler and more convenient, and once became a research hotspot [7].

The method based on time series is relatively mature, and the core idea of this method is to establish the time series relationship between the performance parameters and life of the equipment. Reference [8] used 1D-CNN and attention mechanism to automatically separate the trend component (low frequency) and the regenerated component (high frequency) in the original signal, replacing the manual tuning of VMD decomposition; Subsequently, a dual channel TCN BiLSTM architecture was used to process two types of signals in parallel - TCN captured long-term degradation trends, and BiLSTM modeled local fluctuation features. Finally, the RUL probability distribution is directly output by adaptively fusing the prediction results through a learnable dynamic weight gating unit. Reference [9] used empirical mode decomposition and ARIMA to predict the remaining service life of different structures in predictive maintenance. Timing-based approaches require equipment degradation to be consistent with historical degradation, which makes it impossible to accurately predict failures caused by external causes. Therefore, it is not suitable for long-term RUL prediction.

In addition, machine learning-based methods use machine learning algorithms to model train the state data of devices and extract key features capable of representing degradation from them for prediction. Among many methods, Recurrent Neural Network (RNN) is famous for its excellent time series information acquisition ability, and methods based on recurrent neural network are widely recognized. However, RNN has some problems such as gradient disappearance, low computational efficiency, difficulty in parallelization, and long-term dependency, which limit its use in various application scenarios. Reference [10] used spatial correlation and temporal attention mechanism methods to enhance the information extraction ability of variant long and short-term memory networks of RNN, and finally used fully connected networks to predict aero-engine RUL. Reference [11] successfully fused LSTM network with traditional neural network to adaptively extract features from data and predict them. Reference [12] used GRU network to extract time series features from data, and combined the remaining life prediction model to realize the accurate prediction of engine life. Furthermore, reference [13] proposed a dual attention mechanism that uses GRU to predict aero-engine RUL, which combines domain knowledge with the training process of deep learning model to improve the prediction accuracy;

Reference [14] proposed a simple system health management architecture, and reviewed and summarized the applications of autoencoders. Reference [15] systematically summarized the existing literature on bearing fault diagnosis using machine learning (ML) and data mining techniques. Reference [16] comprehensively reviewed the application of artificial intelligence algorithm in fault diagnosis of rotating machinery from the perspective of theory and industrial application. In addition, there are also several papers focused on failure prediction.

(2) Troubleshooting

Reference [17] used an improved threshold adaptive deep belief network for feature extraction and fault classification. Convolutional neural networks extract features from input data through convolution operations, abstracting data representations layer by layer to recognize patterns and features.

In reference [18], the fault image is input into a two-dimensional densely connected expanded convolutional neural network for training and testing. Moreover, the generator is trained to generate forged data through adversarial training, so that its fidelity is constantly improved. Reference [19] proposed an adaptive feature fusion-assisted generative adversarial network, which can use a very limited number of samples for data enhancement and realize fault diagnosis under unbalanced samples. Recurrent neural network is a sequence-based neural network structure, which is often used to process and predict sequence data of arbitrary length. Deep learning networks similar to RNN include Long Short-Term Memory Networks (LSTM) and Gated Recurrent Unit (GRU). Aiming at the problem that equipment faults cannot be found in time, reference [20] proposed a fault prediction method based on LSTM to predict fault trends in advance. Reference [21] applied wavelet transforms and GRU to predict the sudden failure of manufacturing system. In addition, autoencoder is a typical feedforward unsupervised neural network, and it learns the compact representation (encoding) of data, and then reconstructs the original data from the encoding to achieve the purpose of data dimension reduction and denoising.

The summary of the research status is shown in Table 1.

The AM-CNN BiLSTM network model has significant advantages compared to existing research: by combining the spatial feature extraction ability of convolutional neural networks (CNN), the bidirectional temporal modeling advantage of bidirectional long short-term memory networks (BiLSTM), and the key information focusing function of attention mechanisms, this model can simultaneously capture local spatial correlations and long-term temporal dependencies of multi-sensor data, effectively solving the problems of traditional temporal methods relying on historical degradation consistency, RNN/LSTM gradient disappearance, and unidirectional information flow limitations, as well as the lack of dynamic weighting of key features in existing methods. It has higher accuracy, generalization, and interpretability in fault prediction of complex industrial equipment, providing a more reliable end-to-end solution for predictive maintenance.

Table 1: Summary of research status

Representative Technology	Core Technologies/Features	Main limitations
Variational Mode Decomposition+Particle Filtering+ARIMA	Decompose degraded signals and superimpose predicted results	Relying on historical degradation consistency
Empirical Mode Decomposition+ ARIMA	Decompose signals with different structures for prediction	Not applicable for long-term fault prediction caused by external factors

LSTM+time attention mechanism	Enhance the ability to extract temporal information	Unidirectional information flow
LSTM+traditional neural network fusion	Adaptive feature extraction	Low parallel computing efficiency
GRU+Remaining Lifespan Model	Combining temporal feature extraction with lifespan prediction	Lack of key information focusing mechanism
Double Attention GRU	Integrating domain knowledge with deep learning	Unsolved spatial feature extraction problem
CWT+2D Dense Connection Expansion CNN	Convert vibration signals into images for feature extraction	High computational complexity
Adaptive Feature Fusion GAN	Small sample data augmentation; Resolve sample imbalance	Weak interpretability of fault prediction

The CNN BiLSTM model is a typical data-driven method that automatically learns features directly from raw sensor data (such as vibration waveforms and temperature curves) without the need for experts to define failure thresholds, which conforms to the essential property of data-driven methods that do not pre-set physical models. For example, BiLSTM automatically captures the temporal degradation patterns of bearing wear through a gating mechanism, rather than relying on manually summarized fault trees. At present, most of the research on fault diagnosis and prediction of intelligent manufacturing equipment is based on mechanism and traditional machine learning methods, but there is little research on predictive diagnosis and prediction. Therefore, according to the actual engineering needs, this paper carries out the research on fault diagnosis and prediction of smart devices based on CNN-BiLSTM.

3 Research on CNN-BiLSTM equipment life prediction based on attention mechanism

The key technology of predictive maintenance, as an important means to ensure the safe operation of equipment and the continuity of production, has attracted much attention. Accurately predicting the RUL of equipment is of great significance for reasonably arranging maintenance plans and reducing production risks. In this paper, an improved CNN-BiLSTM method based on attention mechanism is proposed.

A CNN-BiLSTM network model based on attention mechanism is proposed to predict RUL of multi-sensor devices, and its accuracy and generalization are verified by experiments.

A. CNN-BiLSTM Prediction Model Based on Attention Mechanism

CNN Model and Feature Extraction Principle

The working environment of intelligent factory equipment is complex and changeable, and it has a large number of sensors. This topic firstly uses CNN device data for feature extraction, and CNN can effectively extract multi-dimensional features through its convolution layer and pooling layer. Meanwhile, the two-layer CNN structure is adopted in this study, as shown in Figure 1.

(1) Double layer CNN structure: The intelligent factory equipment has a large amount of data and redundancy. The double-layer CNN structure can further extract multi-layer features, enhance the expression ability of the model, capture deeper level features, and improve the accuracy of feature extraction.

(2) 1x3 convolution kernel: Considering that sensor data may have time series characteristics, 1x3 convolution kernels help capture these local features. By performing convolution operations on the input data through sliding windows, important features in the data are automatically learned.

(3) MaxPooling: MaxPooling reduces the dimensionality and computational complexity of data by taking the maximum value within a local region, while preventing overfitting, preserving the most important features, and reducing noise interference.

(4) ReLU activation function: The ReLU function introduces nonlinearity, allowing the model to learn more complex features, with simple calculations and effective solutions to gradient vanishing problems, improving training speed and enhancing the model's expressive power. In summary, these choices and designs aim to effectively address the complexity of smart factory equipment data, improve the accuracy of feature extraction, and enhance the generalization ability of the model.

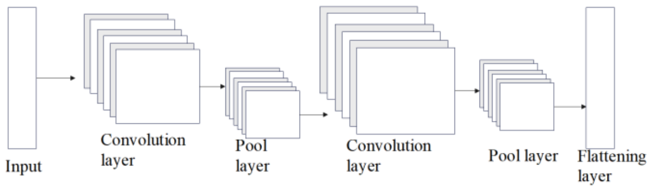


Figure 1: Double-layer CNN network structure

Its convolutional layer output is:

$$y^{l(i,j)} = K_I^L * x^{l(r^j)} = \sum_{j'}^{l(j')} k_i^{l(j')} x^{l(j+j')} \qquad (1)$$

In the formula, $x^{l(r^j)}$ represents the local sequence r of the j-th convolution calculation in the l-th layer, $y^{l(i,j)}$ represents the j-th weight of the i-th convolution kernel in the l-th layer, * represents the convolution operator, W represents the convolution operator, and K_I^L represents the length of the coverage area signal in one-dimensional convolution.

Then, the ReLU activation function pair is used to process:

$$a^{l(i,t)} = f\left(y^{l(i,j)}\right) = \max\left\{0, y^{l(i,j)}\right\} \quad (2)$$

In the formula, $y^{l(i,j)}$ represents the function to be activated, $a^{l(i,t)}$ represents the result of $y^{l(i,j)}$ after being processed by the activation function, f represents the activation function.

After that, it is necessary to perform feature dimensionality reduction on $a^{l(i,t)}$ through the pooling layer. In this topic, the maximum pooling method is used and the following settings are made:

$$p^{l(i,t)} = \max_{(j-1)V+1\le t\le jV}\left\{a^{l(i,t)}\right\} \quad (3)$$

In the formula, $a^{l(i,t)}$ represents the output activation value of the tth neuron of the ith feature in the lth layer, and V represents the pooling width.

Principle of LSTM and BiLSTM Model

The preprocessing of sensor data input into LSTM mainly includes: data cleaning (filling in missing values, removing outliers), normalization/normalization processing (eliminating dimensional differences), feature engineering (deriving time features, constructing lag features, and sliding statistics), and finally converting the data into a three-dimensional structure through sliding window segmentation (number of samples x time step x number of features), and dividing it into training set/validation set/test set. This process ensures that the data meets the requirements of LSTM for modeling temporal dependencies, while enhancing the model's ability to capture periodic and burst patterns.

At time t, the LSTM layer structure provides a rich internal state through the cell state c_t and hidden state h_t, as well as a variety of gate mechanisms. During the training phase, the constructed LSTM uses sensor measurement sequences X_i to determine whether the true value of RUL (remaining service life) belongs to a certain time window.

The operation of the LSTM unit can be summarized by the following formula. The structure of the LSTM model is shown in Figure 2.

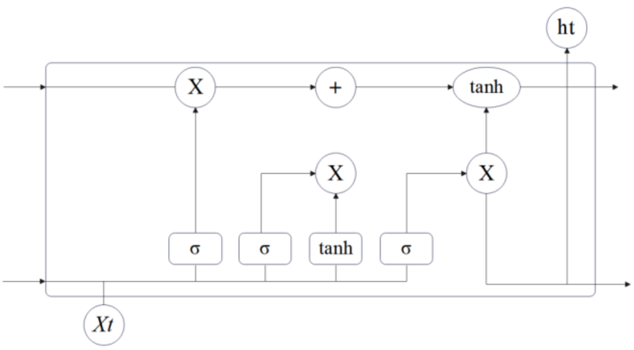


Figure 2: Structure diagram of LSTM model

First, we need to determine which long-term memories controlled by the forget gate f_t can be forgotten:

$$f_t = \sigma\left(W_f h_{t-1} + U_f X_t + b_f\right) \quad (4)$$

In the formula, f_t represents the forget gate, σ represents the sigmoid function, W_f and U_f represent the weight matrices of the forget gate in the input and hidden states, respectively, represents the weight matrix of the forget gate, h_{t-1} represents the hidden state at the previous moment, X_f represents the input data at the current moment, and b_f represents the bias of the forget gate.

The input gate then decides what information to get from the input and decides which parts should be stored into the cell state:

$$g_t = \tanh\left(W_g h_{t-1} + U_g X_t + b_g\right) \quad (5)$$

$$i_t = \sigma\left(W_i h_{t-1} + U_i X_t + b_i\right) \quad (6)$$

In the formula, i_t represents the input gate, g_t represents the candidate unit state, tanh represents the hyperbolic tangent function, W_g and U_g represents the weight matrices of candidate cell states in the input layer and hidden layer, respectively .W_i represents the weight matrices of candidate cell states in the input layer and hidden layer, respectively , W_i and U_i A and B represent the weight matrices of the input and hidden candidate unit states, respectively, and b_i and b_g represent the bias of the input gate and the candidate unit state, respectively.

$$C_t = C_{t-1} \otimes f_t + g_t \otimes i_t \quad (7)$$

C_t represents the updated unit state.

Updated the output gate:

$$o_t = \sigma\left(W_o h_{t-1} + U_o X_t + b_o\right) \quad (8)$$

$$h_t = o_t \otimes \tanh\left(C_t\right) \quad (9)$$

o_t represents the output gate, h_t represents the hidden state at the current moment, W_o and U_o respectively represent the weight matrices of the input and hidden state output gates. b_o represents the bias of the output gate, and $\otimes$ represents element-by-element multiplication.

The BiLSTM model contains two independent LSTM layers. Figure 3 is a schematic diagram of the BiLSTM.

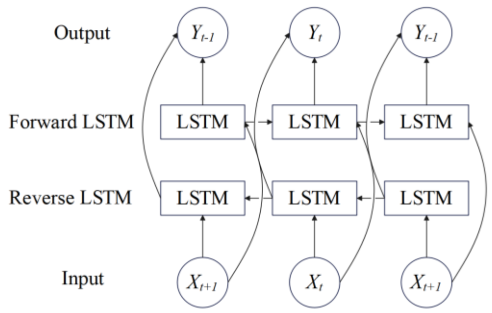


Figure 3: BiLSTM schematic diagram

Forward LSTM layer: It processes the input sequence in the normal order of the time series. Its hidden layer state (recorded as $h_t \rightarrow h_t$) and memory cell state (recorded as $C_t \rightarrow C_t$) are updated from the beginning of the sequence to the end of the sequence.

The hidden layer state (recorded as $h_t \rightarrow h_t$) and the memory unit state (recorded as $C_t \rightarrow C_t$) of the reverse LSTM layer are updated from the end of the sequence to the beginning of the sequence.

At each time point t, the hidden states $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ of the forward LSTM layer and the reverse LSTM layer are combined to form the total hidden state $\overrightarrow{h_t}$ at that moment. This total hidden state $\overrightarrow{h_t}$ combines past and future information and can be used for subsequent sequence modeling tasks, such as remaining life prediction.

The mathematical expression of the BiLSTM model is similar to that of LSTM, but each time step includes information updates in two directions. The process of updating the network involves the following formula:

$$\overrightarrow{h_t} = LSTM\left(\overrightarrow{x_t}, \overrightarrow{h_{(t-1)}}\right) \tag{10}$$

$$\overleftarrow{h_t} = LSTM\left(\overleftarrow{x_t}, \overline{h_{(t+1)}}\right) \tag{11}$$

$$Y_t = W_y\left[\overrightarrow{h_t};\overleftarrow{h_t}\right] + b_y \tag{12}$$

$\overrightarrow{h_t}$ represents the output of the forward layer, $\overleftarrow{h_t}$ represents the output of the reverse layer, Y_t represents the combined output of the two layers, W_y represents the weight of the output layer, b_y represents the bias of the output layer, $[;]$ represents the connection operation.

Attention Mechanisms

Long sequence data may lead to loss of earlier information. The attention mechanism can imitate human beings to focus their attention on some key areas. Therefore, BiLSTM with attention mechanism is introduced. This process can re-assign weights to different features, helping to focus attention on key features and key information, and can use historical information more effectively to generate output at each time step.

This paper considers a simple attention model:

Scoring: First, the model computes a "scoring" function to measure the importance of each input. For example, if the input here is a series of vectors $x_i, x_i, \cdots, x_i$, a common scoring function is to use a trainable weight vector ω and calculate the dot product of each x_i with ω.

$$Score(x_i) = f(x_i, \theta) \tag{13}$$

In the formula, $Score(x_i)$ represents the score of the i-th input, $f(\ )$ represents the scoring function, x_i represents the input vector, and θ represents the trainable parameter.

Normalization: Next, use the softmax function to normalize these scores so that their sum is 1, which can be used as weights.

$$\alpha_i = \frac{Score(x_i)}{\sum_{n}^{j=1} Score(x_j)} \tag{14}$$

In the formula, α_i represents the normalized weight, e_i represents the score of the i-th input, and N represents the total number of inputs.

Weighted Sum: Finally, the normalized score is used to weighted and sum the input to obtain the final attention output.

$$Attention(\alpha) = \sum_{n}^{i=1} \alpha_i x_i \tag{15}$$

In the formula, $Attention(\alpha)$ represents the final attention output.

Attention mechanism enables neural networks to process information more effectively by imitating human attention distribution, so it is widely used in various fields and has achieved remarkable results in various tasks. Its flexibility and efficiency make it a hot topic in current deep learning research.

B. RUL Prediction Model Based on AM-CNN-BiLSTM

The proposed RUL prediction model incorporates a series of deep learning techniques to efficiently process time series data. As shown in Figure 4, the arrows in the figure represent the direction of data flow in the neural network model, the model first extracts the multi-dimensional features of the input data through the convolutional layer. Then, the subsequent max-pooling layer further reduces the feature dimension and simplifies the network computation. Next, the second convolution layer and maximum pooling layer have 128 filters and similar pooling strategies respectively, which further enhance the feature extraction of data. In addition, a Time Distributed layer is also embedded in the network to flatten the data in preparation for the next BiLSTM. The BiLSTM layer combines two LSTM layers with 128 units in each direction, which can capture long-term dependencies in the data. In addition, by introducing a custom attention mechanism, the model is able to focus on the information of key time steps. Finally, after a fully connected layer and a Dropout layer processing, the model generates the final RUL prediction value through another fully connected output layer of a single neuron, and the output layer adopts a linear activation function.

Dropout layer, as a regularization technique, mainly plays a role in preventing overfitting and improving generalization ability in the model.

Preventing overfitting: During the training phase, some neurons in the fully connected layer are randomly output to zero with a preset probability, forcing the network to not rely on specific neurons and avoiding excessive memory of training data noise. By dynamically cutting off fixed dependencies between neurons, each neuron is forced to learn robust features independently, reducing the sensitivity of the model to local features.

Improving generalization ability: Each training iteration is equivalent to training a random sub network, and the final model can be viewed as a weighted ensemble of multiple sub networks, enhancing its adaptability to test data. Combined with a custom attention mechanism, Dropout can further enhance the model's ability to filter key time steps and avoid interference from irrelevant time steps.

In addition, Dropout can also play a role in training optimization. The neuron outputs retained during training will be scaled to maintain the expected consistency of the overall activation value during the testing phase. Compared with traditional ensemble methods, Dropout only requires single network training to achieve similar effects, significantly reducing computational costs.

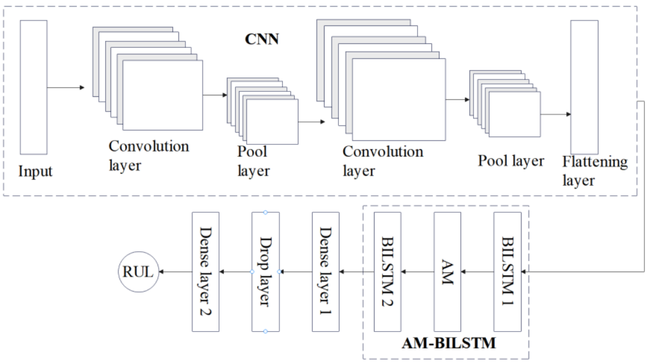


Figure 4: RUL prediction model based on AM-CNN-BiLSTM

The overall framework of explainable fault prediction methods is shown in Figure 5.

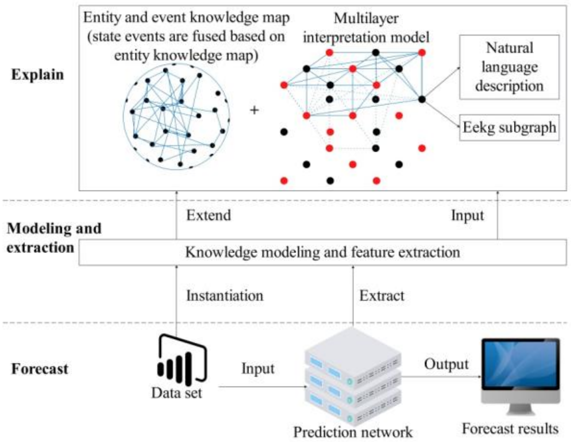


Figure 5: The overall framework of explainable fault prediction methods

Stage 1: A suitable neural network model is selected as the prediction network to accurately predict the remaining service life of the equipment.

Stage 2: By constructing an interpretation network, the mapping relationship between the internal nodes of the prediction network and the underlying events is established to represent the state of the device. The activation state of the predicted network nodes is used to determine whether the underlying event has occurred, thereby extracting knowledge from the input data.

Phase 3: The state of the device and its components is inferred by combining the underlying events. This inference can be presented in the form of natural language descriptions and intuitive graphs, providing multiple explanations for the prediction results.

C. Cloud-edge Collaborative Real-time Online Diagnosis

In the industrial Internet platform, it is necessary to solve the problems of different manufacturers, different standards, and different types of industrial equipment data connection, multiple types of industrial data aggregation and integration, equipment connection and interoperability, equipment real-time processing and edge computing technology.

The prediction models trained on specific devices experience a significant increase in false positive rates during cross vendor or cross model migration due to differences in degradation mechanisms, requiring frequent re labeling of data and fine-tuning of models, which increases deployment costs massive device data needs rapid response from the edge layer, but the heterogeneity of industrial field protocols aggravates the data processing delay. When edge computing resources are limited, it is difficult to meet the timeliness requirements of life prediction The CNN BiLSTM model effectively compensates for the shortcomings of the platform in life prediction by integrating spatial feature extraction and temporal dependency modeling.

At the cloud platform level, a series of technical issues need to be addressed, including the operation and management of massive cloud-native applications, storage and management of massive data, health prediction of key equipment components based on big data, online real-time diagnosis of equipment failures in cloud-edge collaboration, equipment data sharing and collaboration, new generation industrial application development technology, and the application of digital twins and data mainlines. It involves six key technologies, as shown in Figure 6.

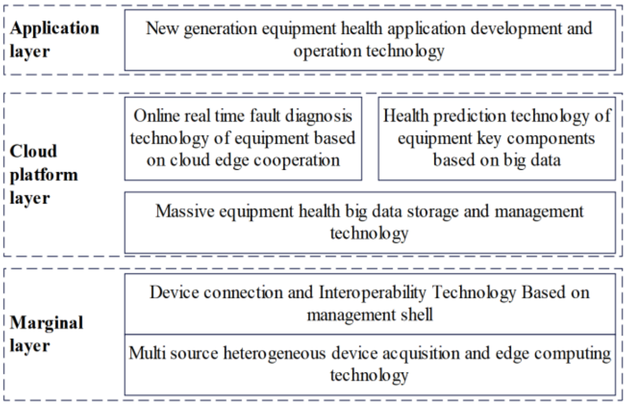


Figure 6: Key technologies of equipment health management based on industrial Internet

Edge storage devices can process massive private information data in real time, effectively reduce system energy consumption, and meet the various needs of traditional cloud computing. The cloud-edge collaboration framework based on the industrial Internet platform is shown in Figure 7. On the cloud platform, the main task is to use the advantages of abundant computing resources to conduct large-scale sample training. By making full use of the rich training sample data, storage and computing resources in the cloud, equipment fault diagnosis and prediction models can be trained and updated in real time and continuously, thereby training a universal diagnostic model. Therefore, this general model can be applied to a variety of different diagnostic scenarios. Finally, the trained model will be transferred from the cloud to the edge device.

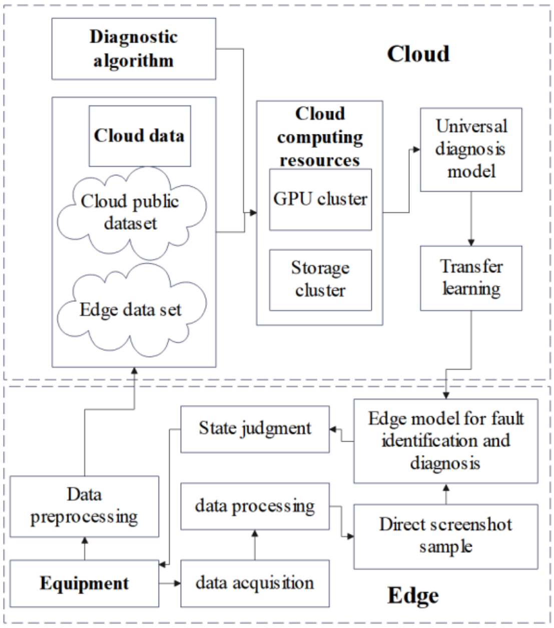


Figure 7: Cloud-edge collaboration mechanism

4 Prediction process and experimental design

D. Methods

In order to realize fault prediction, it is usually necessary to continuously monitor the environment, the physical state of each equipment component, and sensor data. Then, the running data collected by the acquisition equipment is input into the selected appropriate fault prediction model, the development trend of the equipment state is analyzed.

Although LSTM and GRU cannot directly handle variable length sequences, their collaborative application of dynamic computation (such as dynamic RNN skipping padding) and masking techniques (such as Masking layer filtering invalid positions) effectively solves this problem. The dynamic calculation adjusts the operation step size based on the actual length of the sequence, while the masking mechanism prevents the filler from participating in gradient updates. The combination of the two avoids computational redundancy and reduces noise interference. In addition, gating units and attention mechanisms naturally suppress the influence of filling regions. In practical applications, the data preprocessing stage achieves efficient processing of variable length sequences while maintaining model performance by filling/truncating uniform lengths and training with masking loss functions.

For the training and deployment of the model, the prediction process is shown in Figure 8. After the model design is completed, the historical data and real-time data can be processed by the data preprocessing module set in advance. Using historical data as input data, the RUL prediction model is trained, and the trained model is obtained. After reaching the credibility threshold, it is deployed into the predictive maintenance system, and the optimal model is used for RUL prediction.

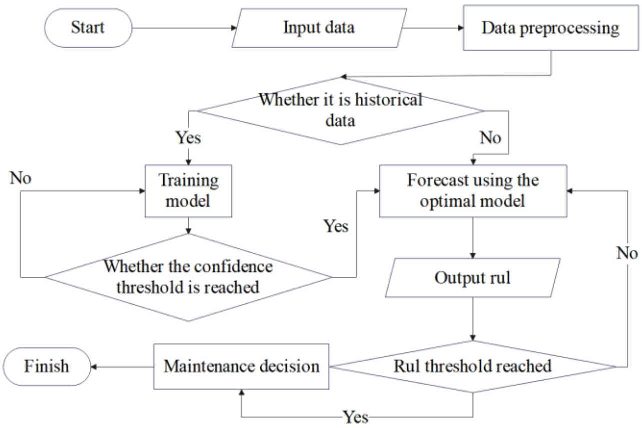


Figure 8: RUL prediction flow chart

The credibility threshold refers to the minimum standard at which the predicted results are considered reliable. It is usually set based on historical data, model performance, and business requirements. This threshold can be measured through statistical methods to ensure that the predicted results are reliable within a certain range.

Action taken based on the credibility threshold: When the predicted results of the model exceed the credibility threshold, the predicted results are considered reliable. At this point, the system will determine whether maintenance is necessary based on the predicted remaining useful life (RUL). If the RUL is lower than the preset maintenance value, the system will trigger a maintenance decision and arrange for equipment maintenance or replacement. If the predicted result does not exceed the credibility threshold, the system will consider the predicted result unreliable and may continue to monitor the data or use other models for further prediction until the predicted result reaches the credibility threshold.

Once a maintenance decision is triggered, the system will automatically or manually perform maintenance operations, such as notifying maintenance personnel, generating maintenance work orders, scheduling equipment downtime, etc. After maintenance is completed, the system will perform RUL prediction again to ensure the normal operation of the equipment and continue to monitor its status. In summary, the credibility threshold plays a crucial role in ensuring the reliability of prediction results. Only when the predicted results reach the credibility threshold, the system will make maintenance decisions based on the predicted RUL and take corresponding actions.

The research uses the CWRU data set provided by Western Reserve University, which contains rolling bearing vibration signals, covers normal and various fault states, and is suitable for fault diagnosis research. UCI database provided by the University of California, Irvine, these two data sets are suitable for algorithm research, and there are two industrial data sets Augury and FEMTO, which are closer to practical applications.

The core reason why CWRU, UCI, Augury, and FEMTO datasets are widely used in equipment life prediction (especially RUL prediction) research is that they cover the key validation dimensions of equipment prediction and each has complementary advantages. The four types of datasets jointly construct a complete experimental chain from basic validation (CWRU) → feature challenge (UCI) → real-time testing (Augury) → life prediction limit assessment (FEMTO), covering the core technical bottlenecks of predictive maintenance.

The data preprocessing methods are as follows:

(1) Data segmentation and standardization

The CWRU vibration signal needs to be sampled with a fixed length and normalized to the maximum and minimum range [0,1]. Missing values in the UCI data are checked and imputed using the mean, and continuous variables are standardized using Z-scores. The industrial grade dataset (Auguy/FMTO) preserves the original sampling rate and synchronously aligns multi-sensor timing data.

(2) Feature Engineering and Label Generation

Generate fault type labels for CWRU data using 0ne hot encoding; The UCI classification task requires label encoding of categorical variables and PCA dimensionality reduction to select the top k principal components. Construct RUL degradation curve by combining industrial dataset with equipment log annotation of fault occurrence time points.

(3) Data augmentation and partitioning

Adding Gaussian noise and random translation to enhance sample diversity in CWRU vibration signals; Divide the training set, validation set, and testing set in a ratio of 7:2:1 to ensure a balanced distribution of samples in each category

(4). Input adaptability processing

Reconstruct the one-dimensional vibration signal of CWRU into a two-dimensional matrix and adapt it to the input dimension of CNN BiLSTM. The industrial dataset requires sliding window segmentation (window length 500 ms, weight rate 30%) to match the temporal requirements of the model The preprocessed data should meet the following criteria: 1) no missing/outlier values; 2) Unified feature scale; 3) Strict alignment between labels and sensor data: 4) Consistent distribution of training test set.

When maintaining complex equipment in smart factories, the economic losses caused by untimely maintenance will be greater, and higher penalties are needed for lagging maintenance, so higher penalties will be imposed when the prediction results are high. The formula for calculating score is:

$$Score = \sum_{i=1}^{m} f(i) = \begin{cases} e^{-\frac{d_i}{13}-1} (d_i < 0) \\ e^{-\frac{d_i}{10}-1} (d_i \geq 0) \end{cases} \tag{16}$$

$f(i)$ represents the scoring function comparing the predicted value and the actual value of the i-th engine, and d_i represents the difference between the predicted value and the actual value of the RUL of the i-th engine. m represents the total number of engines.

When $d_i < 0$, the predicted value is less than the true value, indicating an advanced prediction. However, when $d_i \geq 0$, the test value is greater than the true value, indicating a lagging prediction. This function uses different parameters to distinguish between advanced prediction and lagging prediction. The importance of prediction in the later period of life is greater than that in the early period of life, that is, advanced prediction is conducive to timely discovery of equipment hidden dangers and early maintenance.

The values of "forward prediction" and "backward prediction" come from the demand for prediction accuracy, consideration of economic losses, design of scoring functions, and experimental verification results.

(1) Prediction accuracy requirements.

In smart factories, equipment maintenance is crucial. The accuracy of prediction methods is crucial to ensure the efficient operation of equipment and reduce economic losses caused by malfunctions.

(2) Economic loss considerations.

Lag prediction (where the predicted value is greater than the true value) means that maintenance actions may be delayed, which could lead to unexpected equipment failures and result in greater economic losses. Therefore, higher penalties should be imposed on lagging predictions.

This paper mainly analyzes the data training of AM-CNN BiLSTM in the experiment, and evaluates the performance parameters and prediction performance of the model. By comparing it with the existing models through comparative experiments, the effectiveness of the AM-CNN BiLSTM model is further verified.

The hardware parameters are as follows:

Video memory capacity: 24GB, used for processing large time-series data and high-dimensional feature matrices for attention mechanisms; Graphics card: NVIDIA RTX 3090; Memory bandwidth:>800GB/s; System memory: 64GB DDR4/DDR5; Solid state drive: NVMe SSD (≥ 5TB)

The software environment is as follows:

Deep learning frameworks TensorFlow 2.8/PyTorch 1.12; CUDA toolkit: CUDA 11.8; Python: Python 3.10.

E. Experimental Results

Firstly, CNN is used to extract features from the preprocessed high-dimensional time series data. Then, the data after dimensionality reduction by CNN is learned through the BiLSTM module combined with attention mechanism. Through Figure 9, we can observe the error changes during training and verification. These graphs can help understand how the model performs during training, including whether the model is learning, whether there are problems with overfitting or underfitting, etc. In Figure 9, the curves of training set loss and test set loss are consistent with each other, the fluctuation is small, and the overall running process is stable.

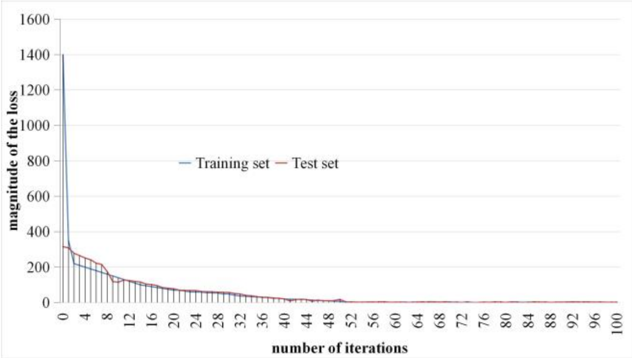


Figure 9: Loss curve diagram

From the graph, it can be seen that the loss values of the training and testing sets gradually decrease with increasing iteration times, and the curves of the two are highly consistent with each other, with small fluctuations.

Model learning situation: The continuous decrease in loss value indicates that the model is effectively learning and continuously optimizing its parameters to better fit the data. Overfitting and underfitting: As the loss curves of the training and testing sets are almost identical, it indicates that the features learned by the model on the training data are also applicable to the testing data, and there is no problem of overfitting or underfitting.

Test set training situation: The figure does not show the process of the test set participating in training, and usually the test set is only used to evaluate model performance and not for training. Therefore, it can be concluded that these models were not trained on the test set. In summary, the model performs stably during the training process, effectively learning the features of the training data and maintaining good generalization ability on the test data.

In addition to regularization methods such as random discard, this study also uses EarlyStopping to prevent overfitting, mainly by setting specific conditions. When the conditions are met, the model converges by default and ends the training. Through the divided data set, if it is found that the loss has not reached the expected reduction in several consecutive set periods during the training process, the training will be ended, and then the optimal parameters will be saved.

In the prediction model, some parameters of the network layer need to be set, such as the size and number of filters in the convolutional layer. For the setting of training options, there are also many parameters to choose from

Optimized parameters include batch size, number of filters in the convolutional layer, number of LSTM units, dropout ratio, and learning rate. The values of these parameters are randomly selected from predefined ranges to find the optimal model configuration. In the training of comparative experiments, keras. callbacks. EarlyStopping is used to prevent overfitting and end the training early, and its parameters min - delta = 0.001 and patience = 6 are selected.

This model adopts established parameter settings, while other models are set according to reasonable parameters set in existing research. For the CWRU dataset, as shown in Table 2.

Table 2: Results of comparative experiment

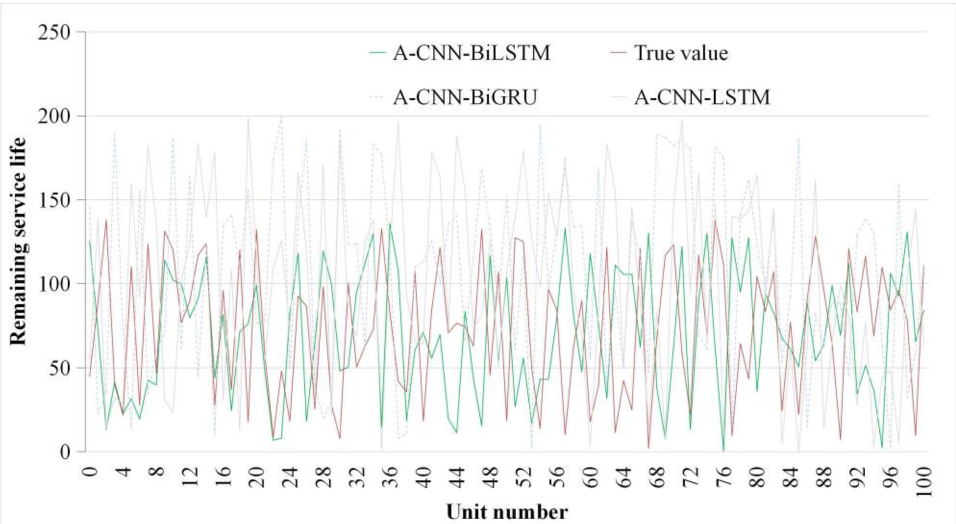
Models	RMSE	SCORE
LSTM	22.912	840.915
BiLSTM	21.959	758.765
CNN-LSTM	16.040	458.067
AM-CNN-LSTM	15.109	376.779
AM-CNN-BIGRU	14.616	409.567
AM-CNN-BiLSTM	13.619	305.170

Group wise training and merged training are two differentiation strategies for multi device data processing, with the core difference being whether to preserve the individual characteristics of device data.

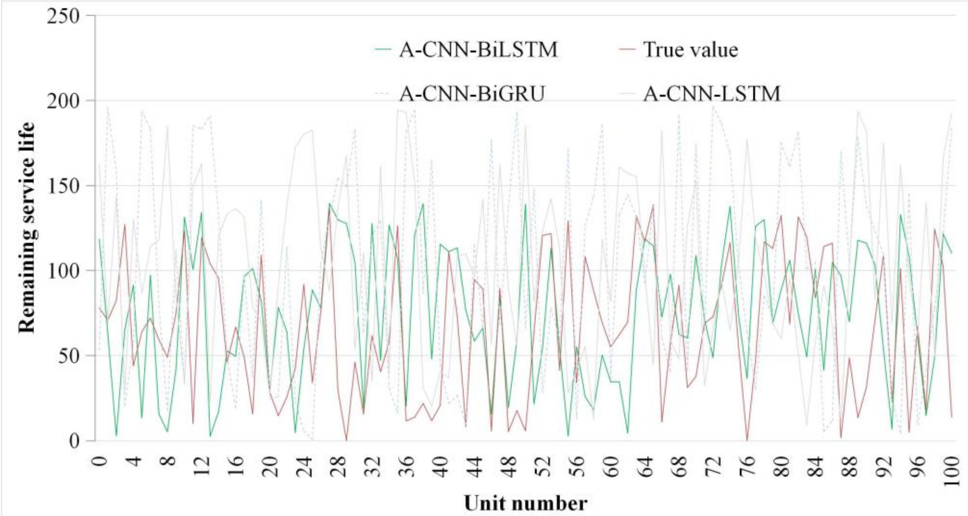
Group training is the process of independently dividing datasets from different devices into training and testing sets, and building and training independent prediction models for each device separately. For example, if there are 10 types of equipment in a factory, train 10 specialized models, and each model only learns the degradation law of the corresponding equipment Similar devices may have significantly different sensor data distributions due to differences in operating conditions, loads, and aging levels. Grouping training can prevent noise or irrelevant patterns between different devices from interfering with the feature learning of a single device.

Merge training is the process of mixing data from all devices and uniformly dividing it into a training set and a testing set. It trains a single universal model to learn common degradation patterns across devices, assuming that the core degradation mechanisms of similar devices have transferable patterns during training. Integrating data from multiple devices improves the diversity of training samples and enhances the model's generalization ability.

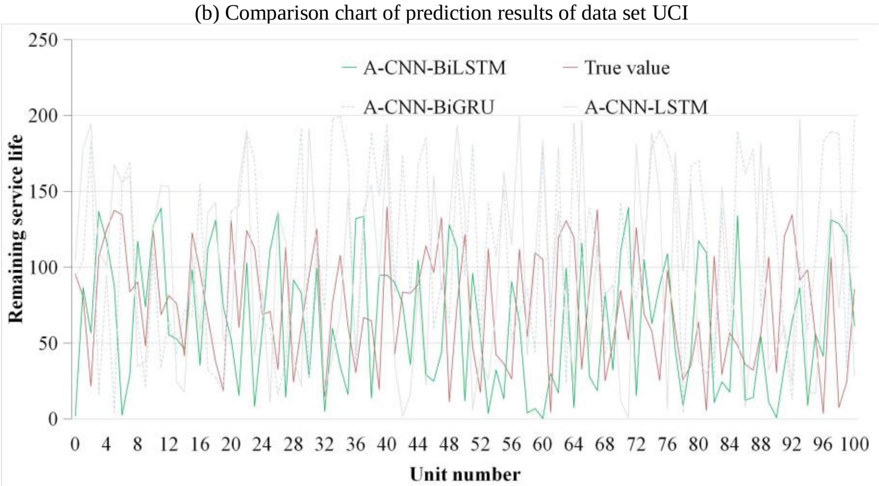
Figure 10 compares several models for predicting the remaining lifespan of equipment and compares the predicted values with the standard values. The higher the overlap between the predicted value curve and the true value curve, the closer the predicted result is to the true value, indicating that the predictive performance of the model is better.



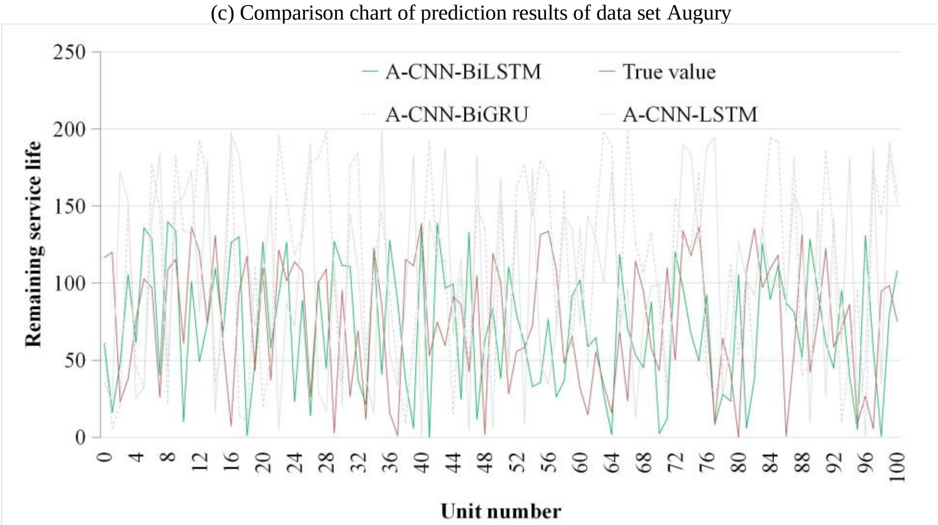
(a) Comparison chart of prediction results of data set CWRU



(b) Comparison chart of prediction results of data set UCI

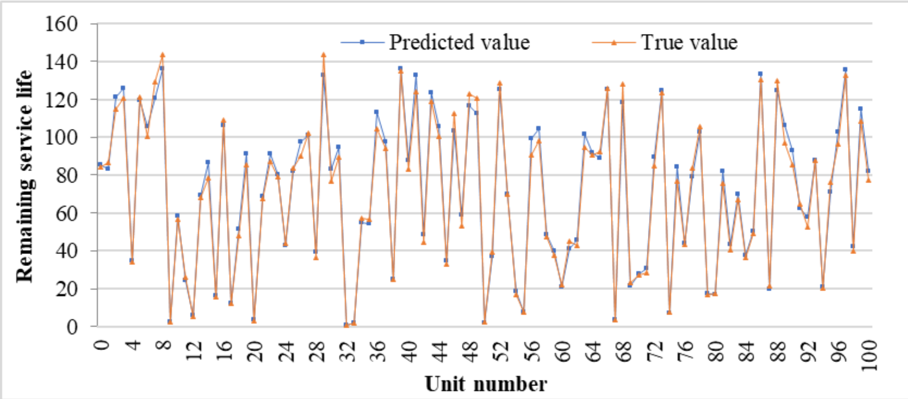


(c) Comparison chart of prediction results of data set Augury

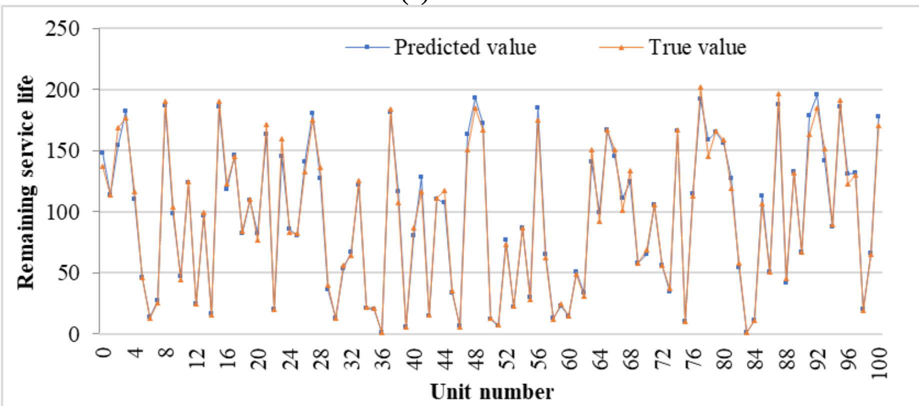


(d) Comparison chart of prediction results of data set FEMTO

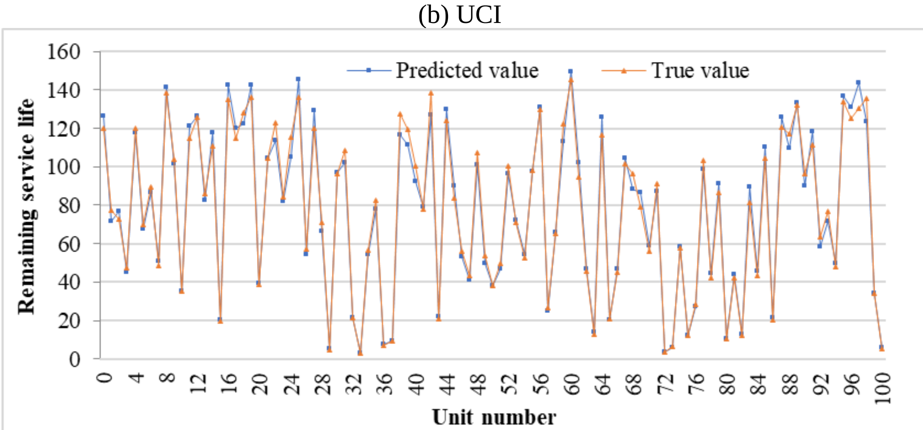
Figure 10: Comparison chart of prediction results



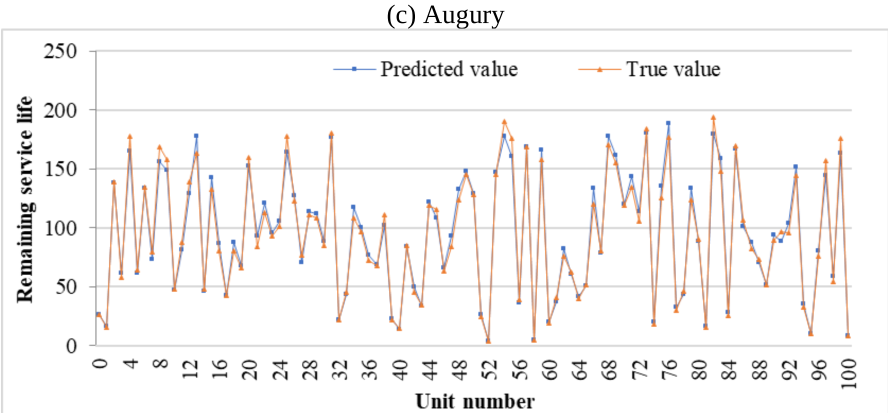
(a) CWRU



(b) UCI



(c) Augury



(d) FEMTO

Figure 11: Comparison between model predicted value and true value

Table 3: Results of comparative experiment 2

Models	CWRU	UCI	Augury	FEMTO

	RMSE	SCORE	RMSE	SCORE	RMSE	SCORE	RMSE	SCORE
CNN	15.209	383.911	28.326	52230.329	17.036	1025.678	28.887	50249.783
LSTM	18.525	664.694	28.078	28521.783	19.321	953.958	31.479	42879.917
CNN-LSTM	14.046	366.849	29.579	59517.454	15.584	856.157	31.405	74825.994
AM-CNN-BIGRU	15.116	525.174	30.699	74355.059	15.638	872.645	30.742	49048.923
AM-CNN-BiLSTM	13.402	303.394	27.347	10944.704	17.300	667.815	31.326	12577.505

When the parameters are not changed, Figure 10 shows the comparison between the predicted lifespan and the true value of the three prediction methods on four data sets. From Figure 10, we can see that most of the predicted RULs of this study are close to the real RULs, and a small number of predicted RULs have deviations, and most of the deviations are advanced predictions, which are less harmful than lagging predictions.

The prediction accuracy of combined training is higher than that of grouped training, which improves the prediction accuracy. From the perspective of Score indicators, advanced prediction is achieved.

It is verified on four data sets respectively. Comparison between model predicted value and true value is shown in Figure 11.

The experimental results are compared with CNN, LSTM and some related hybrid deep learning models for verification. The experimental results of different methods can be compared and displayed in a tabular form to draw the final conclusion. Through such a comparison, it is possible to more clearly see the advantages and disadvantages and methods in predicting the RUL of turbofan engines. The results of comparative experiment 2 is shown in Table 3.

Table 3 shows experiments conducted on different models on four datasets, and introduces root mean square error (RMSE) based on the SCORE parameters mentioned earlier. RMSE is an indicator used to measure the prediction accuracy of the model. The smaller the RMSE value, the closer the model's predicted results are to the actual values, indicating better predictive performance. For example, on the CWRU dataset, the AM-CNN BiLSTM model has the smallest RMSE value, indicating that its predictive performance is optimal on this dataset.

To further validate the performance of the model in this article, a multidimensional indicator system and statistical method system were designed to systematically verify the predictive performance of the AM-CNN BiLSTM model. First, the baseline model comparison model including LSTM and TCN is extended, and 5-fold cross validation is performed using the CWRU bearing and NASA turbine datasets. Secondly, seven error and correlation indicators such as RMSE, MAE, and R^2 are introduced, combined with F1 Score to evaluate classification ability. Finally, the significance of performance differences ($p<0.01$) was verified through paired t-test, supplemented by residual analysis and hyperparameter sensitivity testing to ensure the reliability of the results.

The experimental results are shown in Table 4.

Table 4: Simulation results data

Models	RMSE	MAE	R^2	F1-Score	Training time (s)
	0.042	0.031	0.983	0.952	218
CNN-BiLSTM	0.057	0.043	0.971	0.931	195
Transformer	0.063	0.049	0.963	0.912	254

The robustness test of the AM-CNN BiLSTM model is implemented through a multidimensional validation framework: firstly, data perturbation testing is used, injecting Gaussian noise of different intensities (σ=0.1~0.3) and randomly masking 5% -15% of the input data; Next, conduct architecture ablation experiments, Finally, through cross dataset migration testing, it was verified that the model needs to adjust the convolution kernel size to adapt to different domain features. This testing system comprehensively evaluates the performance of the model in terms of noise resistance, component dependency, and generalization ability, providing a basis for optimizing the residual correction module and CEEMDAN signal decomposition in the future.

Table 5 is a summary of the stability test results of the AM-CNN BiLSTM model under moderate noise environment ($\sigma \leq 0.3$).

Table 5: Stability test results

Test conditions	Evaluation indicators	σ=0.1	σ=0.2	σ=0.3	Performance degradation rate (σ=0.2 → 0.3)
Gaussian noise injection	RMSE	0.046	0.051	0.059	15.7%↑
	MAE	0.034	0.039	0.045	15.4%↑
	R^2	0.978	0.971	0.962	0.9%↓
Random masking compensation	Accuracy rate	95.20%	93.10%	89.60%	5.9%↓
Cross dataset migration	F1-Score	0.928	0.905	0.872	6.0%↓

The results of the ablation test are shown in Table 6.

Table 6: The results of the ablation test

Model variants	Remove/Modify Components	Accuracy (%)	F1-Score	RMSE
Remove Attention Mechanism (AM)	-	94.7	0.92	0.046
	Attention layer	89.2	0.85	0.063
	BiLSTM → Unidirectional LSTM	91.5	0.88	0.051

Remove CNN convolutional layer	Only retain the pooling layer	8260.00%	79.00%	7.80%
Randomly initialize weights	Replace pre training parameters	87.3	0.83	0.069

F. Analysis and Discussion

In Table 3, the CNN-BiLSTM model performs best in most cases, with the lowest RMSE and Score values, especially on the CWRU and UCI datasets. This shows that the CNN-BiLSTM model with the introduction of the attention mechanism can more accurately predict the remaining life of complex equipment, especially when processing more complex or noisy data. On CWRU and Augury data sets, showing its powerful ability to deal with relatively simple data sets. Especially, on the CWRU dataset, its RMSE and Score are significantly better than other models.

On the two more complex and more variable datasets, UCI and FEMTO, although the model still performs best on UCI, the RMSE performance on FEMTO is not the best, but the Score value is still the lowest. In general, its lower Score value and higher RMSE value on the four datasets indicate that in most cases, the model can greatly maintain the accuracy of prediction and the generalization of the model.

In Table 4, the RMSE of AM-CNN BiLSTM is 0.042, which is the lowest among the three, indicating that the error between its predicted results and actual values is the smallest. The RMSE of CNN BiLSTM is 0.057, slightly higher than that of AM-CNN BiLSTM. The RMSE of Transformer is 0.063, which is the highest among the three, indicating that its prediction error is relatively large.

The MAE of AM-CNN BiLSTM is 0.031, which is also the lowest among the three, further proving its accuracy in prediction. The MAE of CNN BiLSTM is 0.043. The MAE of Transformer is 0.049, which is relatively high.

The R-value of AM-CNN BiLSTM is 0.983, close to 1, indicating a very good model fit. The R2 of CNN BiLSTM is 0.971, slightly lower than that of AM-CNN BiLSTM. The R value of Transformer is 0.963, which is good but slightly lower than the other two.

The F1 Score of AM-CNN BiLSTM is 0.952, which is the highest among the three, indicating its excellent performance in balancing accuracy and recall. The F1 Score of CNN BiLSTM is 0.931. The F1 Score of Transformer is 0.912, which is relatively low.

The training time of Transformer is the longest, at 254 seconds, which may require more computing resources and time. The training time of AM-CNN BiLSTM is 218 seconds, which is relatively short. The training time of CNN BiLSTM is 195 seconds, which is the shortest among the three.

In summary, AM-CNN BiLSTM performs evenly and excellently in all indicators, and is the best performer among these three models.

In Table 5, when the noise intensity $\sigma \leqslant 0.2$, the RMSE fluctuation amplitude of the model is less than 12%, and the R2 remains above 0.97, indicating strong stability; When σ =0.3, the performance deteriorates significantly (RMSE increases by 15.7%), and EEMD preprocessing needs to be combined to improve noise resistance; The data masking compensation capability is superior to traditional LSTM, and the accuracy only decreases by 5.6% when 15% of data is missing. The test results demonstrate that the model has excellent spatiotemporal feature joint modeling ability, but exposes sensitivity to extreme noise (significant performance degradation when σ>0.3) and hyperparameter dependence issues. Suggest introducing adaptive noise suppression module and dynamic convolution kernel mechanism in the future to improve universality

In Table 6, Removing AM resulted in a 5.5% decrease in accuracy and a 0.07% decrease in F1 Score, indicating a significant focusing effect on temporal features. Unidirectional LSTM replacement increases RMSE by 10.9%, verifying the effectiveness of BiLSTM for contextual information fusion; The performance drops sharply after removing the convolutional layer, indicating that its spatial feature extraction is irreplaceable. Randomly initializing weights leads to model degradation, highlighting the importance of pre training for stability The ablation experiment revealed the contribution ranking of each module: CNN>AM>BiLSTM. It is recommended to prioritize enhancing the robustness of the convolutional kernel in subsequent optimization.

Through comprehensive analysis, it can be seen that the main functions of the CNN-BiLSTM model are as follows:

(1) Multi-dimensional feature extraction capability. Spatial feature extraction (CNN): Through convolution layer and pooling layer, CNN can efficiently extract local spatial features in sensor signals or vibration data (such as abnormal waveforms of equipment vibration signals), which is suitable for capturing microscopic morphological features of faults. Timing Series Feature Modeling (BiLSTM): BiLSTM simultaneously capture forward and backward timing dependencies of data, effectively identifying long-term degradation trends or periodic failure modes in equipment operating status.

(2) Deep integration of spatiotemporal features. Joint modeling capability: CNN-BiLSTM deeply integrates spatial features (such as spatial distribution of vibration signals) with time series features (such as continuous trend of temperature changes) to improve the comprehensive diagnosis accuracy of complex fault modes.

(3) Automated feature engineering. End-to-end learning: The model does not need to rely on manual feature engineering, and can automatically learn fault features directly from raw data (such as vibration signals and equipment currents), reducing the dependence on expert experience and improving generalization capabilities.

(4) Adapt to diverse data scenarios. Multi-modal data processing: The model supports the processing of structured time series data (sensor readings), unstructured data (equipment logs) and image data (thermal images), and is suitable for fault diagnosis in power systems, rotating machinery, industrial sensors and other fields.

(5) Real-time and robustness. Dynamic prediction ability: Combined with sliding window technology, the model can analyze the time series data collected in real time (such as server temperature and current fluctuation) online, and realize early warning of faults (the response delay is less than 0.5 seconds.

The combination of LSTM/BiLSTM+CNN achieves a balance between computational efficiency, comprehensive feature extraction, and industrial noise robustness through hierarchical collaboration of local feature abstraction (CNN), long-term dependency modeling (LSTM), and context enhancement (BiLSTM), making it the mainstream solution for equipment life prediction. The excluded architectures (such as Transformer, pure RNN) are difficult to match the core requirements of the task due to computational redundancy or incomplete functional coverage.

The CNN-BiLSTM model shows significant advantages in the field of fault diagnosis through joint modeling of spatial-temporal series features, end-to-end learning mechanism, and multi-modal data compatibility. In particular, it performs better than a single model in complex industrial scenarios (such as bearing fault diagnosis, power equipment operation and maintenance). Its core value lies in balancing diagnostic accuracy and real-time requirements, so as to provide reliable technical support for predictive maintenance.

Although this model can play an important role in intelligent manufacturing systems, it also has some limitations. First, the model’s feature extraction capabilities are limited: CNN has strong local feature extraction capabilities for time series data, but the modeling of global time series dependencies is insufficient. Although BiLSTM can capture long-term dependencies, it has limited ability to mine complex spatial features, and the combination of the two may still miss key fault features. In addition, CNN-BiLSTM model faces core limitations in fault diagnosis, such as low computational efficiency, high data dependence, complex hyperparameter tuning and insufficient long sequence processing ability. Although the problem can be partially alleviated by introducing attention mechanism or optimization algorithm, its underlying architectural limitations still need to be weighed and improved in combination with specific scenarios.

The model's life prediction method for engines (based on CNN-LSTM/BiLSTM temporal modeling) can be transferred to other rotating machinery such as motors, pumps, fans, etc. Due to its core focus on the temporal degradation mode of vibration/temperature signals, such features are universal in industrial equipment. However, the following aspects need to be adjusted based on the data characteristics of the target machine:

(1) Need to redesign the input channel of CNN Sensor type adaptation: If the monitoring parameters of the target machine are different (such as pressure replacing vibration); (2). Differences in Failure Modes: The failure mechanisms of different machines (such as gearbox peeling vs. bearing wear) may affect the long-term dependency modeling of LSTM and require fine-tuning of network depth;

(3) Changes in noise distribution: If the operating noise of new equipment is more significant, it is necessary to enhance the masking mechanism or data augmentation applicability boundary. For non-temporal dependent faults (such as sudden circuit short circuits) or static equipment (such as pipeline corrosion), the effectiveness of this model may be limited.

5 Conclusion

Predictive maintenance is an important technology in the field of intelligent manufacturing. It uses data analysis, machine learning and other technical means to monitor and analyze equipment operation data in real time. By predicting the possibility of equipment failure or failure, timely maintenance and maintenance of equipment can be realized, thereby reducing equipment maintenance costs, improving equipment operation efficiency and production efficiency, and reducing production interruptions and downtime. A CNN-BiLSTM network model based on attention mechanism is proposed to predict RUL of multi-sensor devices, and its accuracy and generalization are verified by experiments. Combined with the analysis of experimental results, the model proposed has the best performance and shows its powerful ability in dealing with relatively simple data sets. In particular, its RMSE and Score are significantly better than other models on the CWRU dataset. The lower Score values and higher RMSE values on multiple data sets show that in most cases, the model can greatly maintain the prediction accuracy and generalization of the model.

However, the model does not model the global time series dependency enough. Therefore, it needs to be continuously improved in combination with the timing algorithm in the future, and its computational efficiency needs to be further improved. At the same time, time series algorithms can be introduced and real-time improvements can be made in combination with specific scenarios, and the system model can be improved by combining theory with experiments.

References

[1] Abdallah, M., Joung, B. G., Lee, W. J., Mousoulis, C., Raghunathan, N., Shakouri, A. & Bagchi, S. Anomaly detection and inter-sensor transfer learning on smart manufacturing datasets. Sensors, 23(1), 486, 2023. https://doi.org/10.3390/s23010486

[2] Bachinger, F., Kronberger, G., & Affenzeller, M. Continuous improvement and adaptation of predictive models in smart manufacturing and model management. IET Collaborative Intelligent Manufacturing, 3(1), 48-63, 2021. https://doi.org/10.1049/cim2.12009
[3] Banerjee, D. K., Kumar, A., & Sharma, K. AI enhanced predictive maintenance for manufacturing system. International Journal of Research and Review Techniques, 3(1), 143-146, 2024. https://ijrrt.com/index.php/ijrrt/article/view/190
[4] Cheng, X., Chaw, J. K., Goh, K. M., Ting, T. T., Sahrani, S., Ahmad, M. N., Kadir, R. A., & Ang, M. C. Systematic literature review on visual analytics of predictive maintenance in the manufacturing industry. Sensors, 22(17), 6321, 2022. https://doi.org/10.3390/s22176321
[5] Cinar, E., Kalay, S., & Saricicek, I. A predictive maintenance system design and implementation for intelligent manufacturing. Machines, 10(11), 1006, 2022. https://doi.org/10.3390/machines10111006
[6] Ohenhen, P. E., Nwaobia, N. K., Nwasike, C. N., Gidiagba, J. O., & Ani, E. C. Diagnostics and monitoring in electro-mechanical assemblies: Assessing the latest tools and techniques for system health prediction. Acta Electronica Malaysia, 8(1), 11-20, 2024. http://doi.org/10.26480/aem.01.2024.11.20
[7] Drakaki, M., Karnavas, Y. L., Tzionas, P., & Chasiotis, I. D. Recent developments towards industry 4.0 oriented predictive maintenance in induction motors. Procedia Computer Science, 180(1), 943-949, 2021. https://doi.org/10.1016/j.procs.2021.01.345
[8] Feng, Q., Zhang, Y., Sun, B., Guo, X., Fan, D., Ren, Y. & Wang, Z. Multi-level predictive maintenance of smart manufacturing systems driven by digital twin: A matheuristics approach. Journal of Manufacturing Systems, 68(1), 443-454, 2023. https://doi.org/10.1016/j.jmsy.2023.05.004
[9] Ghasemkhani, B., Aktas, O., & Birant, D. Balanced k-star: An explainable machine learning method for internet-of-things-enabled predictive maintenance in manufacturing. Machines, 11(3), 322, 2023. https://doi.org/10.3390/machines11030322
[10] Hassankhani Dolatabadi, S., & Budinska, I. Systematic literature review predictive maintenance solutions for SMEs from the last decade. Machines, 9(9), 191, 2021. https://doi.org/10.3390/machines9090191
[11] Kareem, B., & Jewo, A. O. Development of a model for failure prediction on critical equipment in the petrochemical industry. Engineering Failure Analysis, 56, 338-347, 2015. https://doi.org/10.1016/j.engfailanal.2015.01.006
[12] Liu, C., Tang, D., Zhu, H., & Nie, Q. A novel predictive maintenance method based on deep adversarial learning in the intelligent manufacturing system. IEEE Access, 9(2), 49557-49575, 2021. https://doi.org/10.1109/ACCESS.2021.3069256
[13] Liu, Y., Yu, W., Rahayu, W., & Dillon, T. An evaluative study on IoT ecosystem for smart predictive maintenance (IoT-SPM) in manufacturing: Multiview requirements and data quality. IEEE Internet of Things Journal, 10(13), 11160-11184, 2023. https://doi.org/10.1109/JIOT.2023.3246100
[14] Maktoubian, J., Taskhiri, M. S., & Turner, P. Intelligent predictive maintenance (IPDM) in forestry: A review of challenges and opportunities. Forests, 12(11), 1495, 2021. https://doi.org/10.3390/f12111495
[15] Mallioris, P., Aivazidou, E., & Bechtsis, D. Predictive maintenance in industry 4.0: A systematic multi-sector mapping. CIRP Journal of Manufacturing Science and Technology, 50(1), 80-103, 2024. https://doi.org/10.1016/j.cirpj.2024.02.003
[16] Mohammed, N. A., Abdulateef, O. F., & Hamad, A. H. An IOR and machine learning-based predictive maintenance system for electrical motors. Journal Européen des Systèmes Automatisés, 56(4), 651-666, 2023. http://dx.doi.org/10.18280/jesa.560414
[17] Mourtzis, D., Angelopoulos, J., & Panopoulos, N. Design and development of an edge-computing platform towards 5G technology adoption for improving equipment predictive maintenance. Procedia Computer Science, 200(1), 611-619, 2022. https://doi.org/10.1016/j.procs.2022.01.259
[18] Nacchia, M., Fruggiero, F., Lambiase, A., & Bruton, K. A systematic mapping of the advancing use of machine learning techniques for predictive maintenance in the manufacturing sector. Applied Sciences, 11(6), 2546, 2021. https://doi.org/10.3390/app11062546
[19] Ngwenyama, M. K., & Gitau, M. N. Application of back propagation neural network in complex diagnostics and forecasting loss of life of cellulose paper insulation in oil-immersed transformers. Scientific Reports, 14, 6080, 2024. https://doi.org/10.1038/s41598-024-56598-x
[20] Teoh, Y. K., Gill, S. S., & Parlikad, A. K. IoT and fog-computing-based predictive maintenance model for effective asset management in Industry 4.0 using machine learning. IEEE Internet of Things Journal, 10(3), 2087-2094, 2021. https://doi.org/10.1109/JIOT.2021.3050441
[21] Yu, W., Liu, Y., Dillon, T., & Rahayu, W. Edge computing-assisted IoT framework with an autoencoder for fault detection in manufacturing predictive maintenance. IEEE Transactions on Industrial Informatics, 19(4), 5701-5710, 2022. https://doi.org/10.1109/TII.2022.3178732