Retracted: "Research on Advanced Prediction of Major Equipment Fatigue Load Based on Deep Learning"

Bowen Yang[1], Chenxu Yang[1], Junzhou Huo[1], Xiangfeng Si[1]
[1]School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China

Abstract： The fatigue life of complex service structures is predicted in real-time according to dynamic random loads. Real-time monitoring of stress/strain in structure key parts is important for practical engineering. Based on the deep learning theory, this paper proposes a long and short-term memory convolutional neural network (CNN- LSTM) prediction model with attention mechanism by using the actual workload data set of key components in major equipment. Firstly, convolutional neural networks (CNN) is used to extract the high-dimensional abstract representation from the preprocessed data, and attention mechanism is added to give each channel different attention scores. Then, long and short-term memory (LSTM) is used to process the time series data and output the final prediction results. In this study, data points in the future are predicted, and the final predicted results are root mean square error (RMSE) of 33.2626 and mean absolute percentage error (MAPE) of 15.3768, which can well fit the future load change trend. At the same time, the multi-condition identification module is added to the code, which can better adapt to the changing load of mechanical equipment in the actual service process. Finally, this paper also designed an ablation experiment, which proved the effectiveness of the overall structure of the prediction model designed in this study and the necessity of the three parts of CNN, ECA-Net and LSTM. It provides theoretical support and practical engineering application for real-time monitoring and predicting the fatigue life of important equipment.

Keywords: Advanced prediction; Deep Learning; Fatigue Load; intelligent monitoring.

1 Introduction

As an important manifestation of national economic development, major equipment is the foundation and core of a country industrial development. Its development is related to the national comprehensive strength and international status [1]. However, as major equipment is increasingly integrated, systematic, precision, equipment structure is increasingly complex, the difficulty of testing and maintenance tasks continue to increase [2]. Mechanical major equipment working situation is complex, poor working conditions, service core key parts subjected to complex alternating loads can easily cause fatigue fracture affecting equipment life, leading to safety hazards and even accidents, bringing significant property losses and casualties [3]. In order to protect the production and processing of normal, continuous, must be prepared in advance spare parts to avoid downtime caused by economic losses, so reliable mechanical life prediction has become an urgent problem in the development of industrial economy [4], as shown in Figure 1. In the current stage of life prediction methods, the method of life prediction based on fatigue load spectrum is a commonly used method in engineering [5]. Accurate and stable acquisition and prediction of core part loads is an important prerequisite for ensuring stable service of major equipment and predicting equipment life [6].

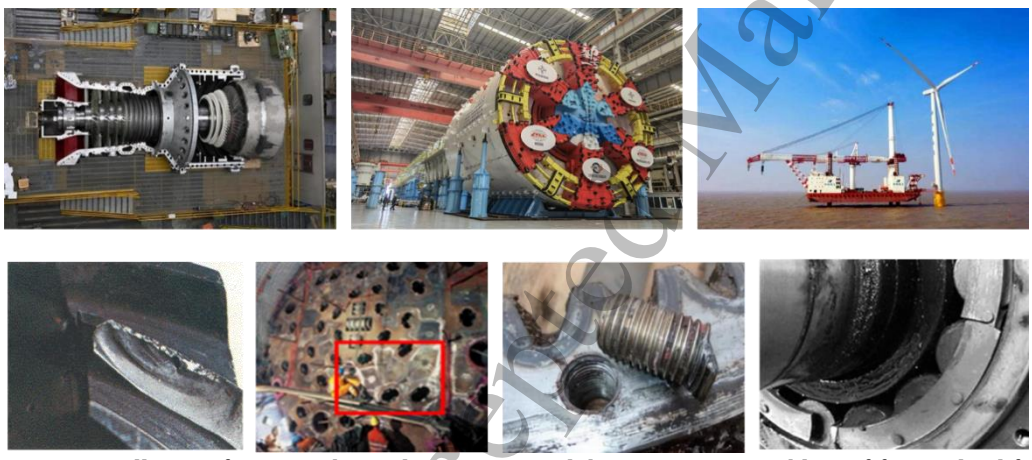


Figure 1 Nationally significant mechanical equipment and the engineering problem of fatigue load failure

With the rapid development of science and technology, the scale and magnitude of data acquired by the modern information society are getting larger and larger, and the accuracy requirements for the prediction of future events are getting higher and higher. The traditional time series regression analysis prediction method is computationally large and has poor generalization ability, especially for the nonlinear relationship prediction effect is not ideal [7]. With the increase of data volume, it is not only more difficult but also difficult to meet the prediction accuracy. Traditional machine learning methods use feature engineering modeling to predict future values, which makes up for the lack of traditional methods that are difficult to capture nonlinear relationships, but because they do not take into account the correlation between the variables of the sequence data, it is difficult to predict with high accuracy as required by modern society [8]. Therefore, the traditional parametric models and machine learning algorithms, which require a lot of theoretical research and experimental data, are not only relatively time-consuming and labor-intensive, but also bring great difficulty to the reliable analysis of industrial processes when there is a complex nonlinear coupling between the relevant variables, which makes it difficult to efficiently and accurately analyze and predict time series data.

Traditional time series forecasting methods are mainly based on statistical methods, and the common ones are AR and MA models as well as ARIMA (Differential Autoregressive Moving Average Model) and Prophet, etc. In 2012, Voyant [9] et al. used a hybrid ARMA/ANN model and data emanating from a numerical weather prediction model (NWP) for forecasting the global radiation. in 2017. Klepsch [10] et al. proposed an approximate vector ARMA model based on generalized principal component analysis (PCA) and investigated the structure of the approximate vector ARMA model.In 2019, Yao Wang [11] et al. used an ARMA model to forecast the average closing price of the SSE balance, and the prediction results differed from the actual value with small differences, and the prediction reliability was high. However, the traditional time series prediction method has a large performance overhead, poor data utilization and generalization ability is not ideal for nonlinear data, and it is difficult to ensure the accuracy of the prediction method.

Machine learning models are computationally fast, the model accuracy is high and missing values do not need to be processed, which is more convenient. Maria [12] et al. used a combination of empirical mode decomposition

and SVM to solve the problem of residual life prediction of non-smooth sequences. In 2015, Gao Yang [13] et al. proposed a machine learning based adaptive prediction model by using the established SVM classifier in combination with ARMA, etc. for the PV output prediction, the performance results are better than the separate ARMA and ANN models.2022, Xue [14] embedded the GA algorithm into the PSO algorithm for parameter optimization, and achieved better results in predicting deep foundation pit deformation of soil-rock composite strata using the GA-PSO-GLSSVM model. However, the feature extraction work is complicated, the data fitting ability is limited, and it is difficult to deal with large-scale data [15] etc., and the level of feature engineering ability often determines the upper limit of the performance of the machine learning model.

Deep learning-based time series prediction mainly relies on multiple neural network structures to fulfill the prediction task. At the end of the last century, many foreign researchers began to use BP neural networks to predict time series, and in 2022, Liu [16] et al. predicted time series by BP neural networks. 2012, Ding Ming [17] et al. proposed a short-term prediction model for photovoltaic power generation system output power based on the improvement of BP neural networks, and verified the effectiveness of the proposed model and algorithm through the analysis of the prediction results. the effectiveness of the proposed model and algorithm. Although BP neural networks can predict time series, they cannot fully utilize the features of time series due to their own network structure [18].LSTM neural network realizes long-term memory through cell units, and it has been proved that the model is more effective for long-time series processing and prediction. In 2018, Wang [19] et al. proposed a multilayer grid search-based of LSTM prediction model parameter optimization algorithm.In 2019, Yang Qing [20] et al. constructed a deep LSTM neural network and applied it to the global stock index prediction study, demonstrating the broad application prospects of LSTM in the future in the direction of financial prediction and other directions. In 2020, Jun [21] et al. proposed an LSTM with a transformation mechanism in the encoder to process the input information flow and learn the variation information, which demonstrated better performance on five different domain datasets.

Convolutional Neural Networks (CNNs) are a class of feedforward neural networks that contain convolutional computation and have a deep structure, and more researches have started to combine them to obtain better results. In 2020, Xie [22] et al. proposed a trajectory prediction method based on sequential modeling, which fuses CNN spatial extension and LSTM temporal extension to predict the trajectories of surrounding vehicles, and the results satisfy the prediction requirements, providing an effective method for the safe operation of unmanned systems. In the same year, they proposed a traffic collision risk prediction model using long and short-term memory convolutional neural network (LSTM-CNN), and the experiments showed that it was better than XGBoost, LSTM and other models in terms of sensitivity and false alarm rate. The operational efficiency as well as accuracy of model processing can be improved by introducing an attention mechanism, which in turn improves the accuracy of time series prediction. In 2020, Wang [23] et al. implemented a recurrent neural network with an attention mechanism to model the long-term correlation of time series data, and proposed a new attention mechanism to select the relevant time series. In 2020, Du Shengdong [24] et al. proposed a sequence-to-sequence spatio-temporal attention deep learning framework based on convolutional LSTM coding layer and LSTM decoding layer with the aid of an attention mechanism. In 2022, Gou [25] et al. proposed a new attention prediction method based on CNN and Transformer, called ACT-Net. the effectiveness of the proposed ACT-Net was demonstrated through experiments. In 2023, Wu [26] et al. proposed a CNN-GRU ship traffic flow prediction model based on the attention mechanism, and the experimental results show that the proposed model has higher prediction accuracy in the prediction of different traffic flow parameters compared with LSTM, CNN+GRU, etc.

In summary, the research of major equipment as the foundation and core of China's national industrial development, its development depends on China's national comprehensive strength and international competitiveness. At present, the research on the performance and life of major equipment is a major problem that restricts the rapid and stable development of China's economy, so it is of great social significance to ensure the safe and stable service of major equipment, and to accurately predict the trend of load change of mechanical major equipment. By utilizing the powerful feature extraction and nonlinear fitting ability of deep learning, the prediction accuracy can be greatly improved to prevent accidents caused by overloading and provide key data for life prediction. Combined with the mature deep learning algorithm model, the study of how to accurately and stably predict the load of the key core parts, and then ensure the stable service of the major equipment, can provide a guarantee for the development of China's comprehensive national strength and social stability.

2 Theoretical methods

In this paper, the study of fatigue load over prediction of major equipment based on deep learning, firstly, preprocess the input actual working load data, then design the structure of the prediction model, and then carry out the experimental configuration of the prediction model, so as to complete the construction of the overall prediction

Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems. Received July 18, 2023;
Accepted manuscript posted June 06, 2024. doi:10.1115/1.4065819

model. Finally, the prediction model is actually run to analyze the accuracy and final prediction results and design the ablation experiment for comparison, and the specific flow structure of the study is shown in Figure 2.

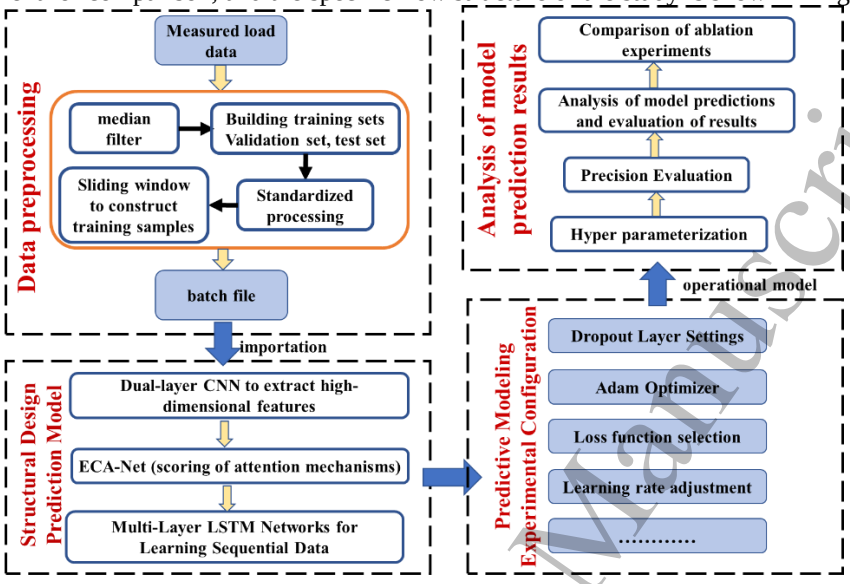


Figure 2 Flow chart of load prediction study

By learning the laws and features in the historical load data of major equipment, it is possible to predict future load trends, which can effectively prevent accidents caused by overload and provide key data for life prediction, and there are many models in deep learning that can be used in load data prediction.

2.1 Long Short-Term Memory Neural Network (LSTM)

Long Short-Term Memory (LSTM) is a variant of Recurrent Neural Network (RNN), due to the unique structural design that makes the LSTM model can effectively solve the gradient disappearance and gradient explosion problems that tend to occur in the RNN, and is suitable for processing and predicting the time series data with long intervals and delays. LSTM unit structure as shown in Figure 3, where x_t is the input of the current time step, h_{t-1} is the output of the previous time, C_{t-1} is the cell state of the input of the previous time, and C_t and h_t are the cell state as well as the output of the current time step, respectively. In addition, σ and tanh represent the sigmoid, tanh activation functions, respectively. LSTM has a chained form of repeating neural network modules, which reads and analyzes the time-series data in a chained fashion. In this study, the LSTM outputs are finally passed through a fully connected layer to arrive at the predicted values.

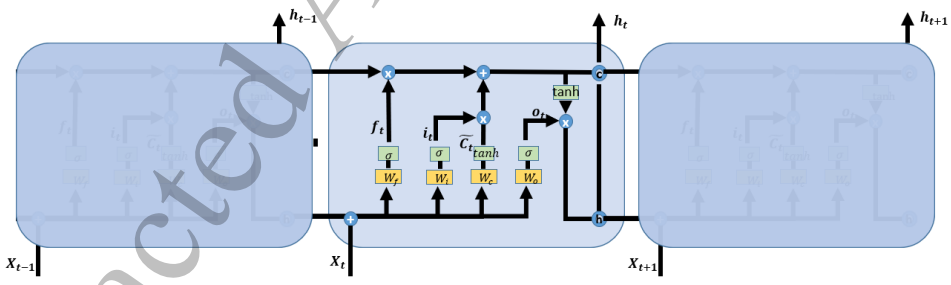


Figure 3 Chain structure of LSTM network

The first step of the LSTM computation is to compute what information to forget from the previous cell state by means of a forgetting gate. In this step, the forgetting gate reads the output h_{t-1} of the previous moment and the temporal input x_t of the current moment, and outputs a vector through the nonlinear computation of the Sigmoid (the values through the σ-operation are all in the range of 0 to 1, the closer to 1 means the more information is retained, and the closer to 0 means the more information is to be forgotten), which is finally multiplied with the cell state of the previous moment. The forgetting gate is calculated as shown in Eq. 1.

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

The second step decides how the new information will be stored in the cell state through the input gate, which consists of two main computational steps: 1. The output h_{t-1} of the previous moment and the input x_t of the current

moment, which is used to derive i_t through the Sigmoid layer (which differs from that in the first step in the difference between the network parameters W and b), decides how much information is to be inputted into the new cell unit; 2. The tanh layer creates a new vector of candidate values $\widetilde{C_t}$ and finally decides how much information to input into the cell state by operation with i_t . The input gate formulas are shown in 2 and 3.

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

$$\widetilde{C_t} = tanh(W_c \cdot [h_{t-1},\ x_t] + b_C) \tag{3}$$

After that, the update of the cell state is carried out, i.e., updating C_{t-1} to C_t. The specific calculation process is as follows: the old state C_{t-1} is multiplied by the output of the oblivion gate, f_t, and then added to the input gate, $i_t * \widetilde{C_t}$, which results in the new value of cell, C_t. The specific calculation process is shown in Eq. 4.

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \tag{4}$$

Finally, we determine the output value h_t for the current time through an output gate. Similarly to before, a sigmoid layer operation is performed to derive the vector o_t, then the cell state is processed through tanh (to get a value between -1 and 1) and multiplied by o_t to get the final output h_t. The output gate formulas are shown in 5 and 6.

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

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

2.2 Convolutional Neural Network (CNN)

The specific process is shown in Figure 4, which demonstrates the process of 2-layer CNN model to deal with the image problem, for the input to first perform the primary convolution and pooling operation and then use a new filter to perform the secondary convolution and pooling operation, and finally unfold the pixels, and fully connected with the fully connected layer to get the final output. The process of using CNN for time series problems is similar, except that the convolution kernel used is a one-dimensional convolution, which is computed by moving only in the horizontal direction and not in the vertical direction.

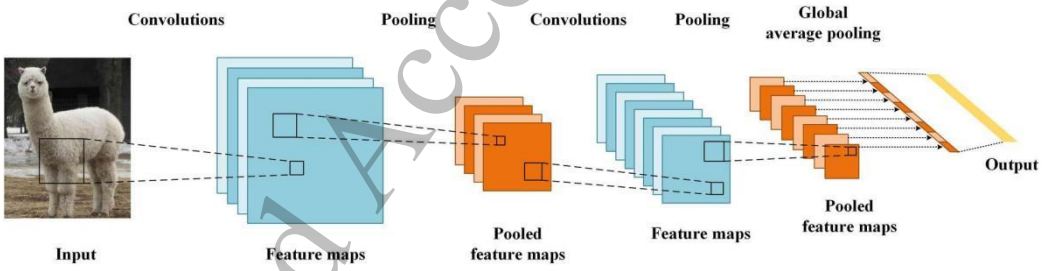


Figure 4 Flowchart of CNN network for processing images

2.3 Attention Mechanism (AM)

The computational process of the attention mechanism is as follows: firstly, the input information is represented in the form of key-value pair, i.e., the input information is $(K, V) = [(k_1, v_1), (k_2, v_2), \cdots, (k_N, v_N)]$, where the "key" (K) is used to calculate the attention distribution α_i and "value" (V) is used to calculate the attention score. The calculation of the attention mechanism can be divided into two parts: the first part is to analyze the input data according to the data Query to be obtained and assign different attention weights to it. The second part is to assign and sum the inputs according to the obtained attention weights to get the final output. Thus, the effect of focusing on important information and ignoring irrelevant information is achieved. The working principle is shown in Figure 5.

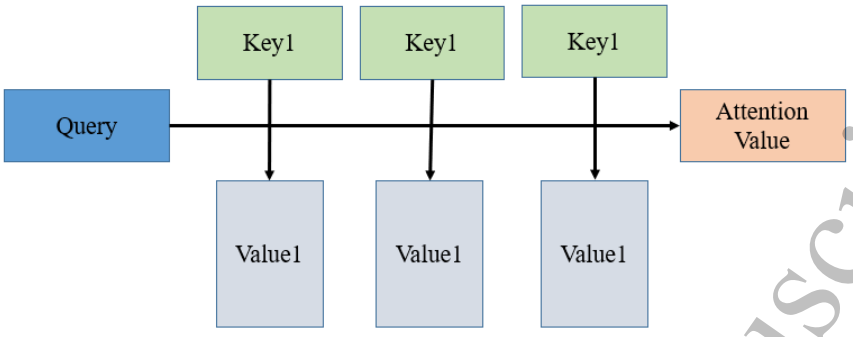


Figure 5 Flowchart of attention mechanism calculation

Where the specific calculation process is accomplished through the following three steps.

Step 1: Calculate the similarity between the two based on Query and Key. It can be calculated by additive model, dot product model, etc. The additive model used in Eq. 7 to get the attention score s_i (where W, U and v are learnable network model parameters).

$$s_i = F(Q, k_i) = v^T \tanh(Wx_i + Uq) \tag{7}$$

Step 2: Numerical conversion of attention scores with softmax function. On the one hand, it allows normalization, converting each result to a value of 0-1 and summing to 1. On the other hand, it highlights the weights of the important elements.

$$s_i = softmax(s_i) = \frac{\exp(s_i)}{\sum_{j=1}^{N} \exp(s_j)} \tag{8}$$

Step 3: Sum the Value weighted according to the weighting coefficients.

$$Attention((K, V), Q) = \sum_{i=1}^{N} \alpha_i v_i \tag{9}$$

Through the calculation of the above three stages, the Attention value for Query can be calculated, thus realizing the different task requirements for different inputs to assign different attention weights, and improving the processing efficiency and accuracy of the task.

The specific formula for each evaluation index is shown in Table 1. where m is the number of samples, y_i is the true value, $\hat{y}_i$ is the predicted value of the model, and $\bar{y}$ is the mean value of y. The model is also known as the goodness-of-fit, which takes the value of [0, 1], and its calculation result is the accuracy of the model prediction.

Table 1 Calculation formula for evaluation indicators

Evaluation indicators	Formula
Mean Absolute Error (MAE)	$MAE(y, \hat{y}) = \frac{1}{m}\sum_{i=1}^{m}(\lvert y_i - \hat{y}_i \rvert)$
Mean Square Error (MSE)	$MSE(y, \hat{y}) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$
Root Mean Square Error (RMSE)	$RMSE(y, \hat{y}) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2)}$
Mean Absolute Percentage Error (MAPE)	$MAPE(y, \hat{y}) = \frac{100\%}{m}\sum_{i=1}^{m}\left\lvert\frac{y_i - \hat{y}_i}{y_i}\right\rvert$
Coefficient of determination R^2	$R^2(y, \hat{y}) = 1 - \frac{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\frac{1}{m}\sum_{i=1}^{m}(y_i - \bar{y})^2} = 1 - \frac{MSE}{Var}$

Each of the five common evaluation indexes mentioned above has its own characteristics, and researchers often use different evaluation indexes to prove the advancement of the algorithms proposed by various algorithms

for the prediction task. In this study, the data of anomalies are more sensitive and there are no data points with the value of 0, so we choose to use RMSE and MAPE as the evaluation indexes.

3 Experimental treatments

Strain monitoring is applied to the TBM construction site of a bid section of "Xinjiang EH Water Diversion and Supply Project". The TBM tunneling process is complicated due to the tunneling parameters and the geology of the bid section. Its main load-bearing components bear complex random loads for a long time. The wireless real-time monitoring sensor shall be installed at the bearing part of TBM cutter head, as shown in Figure 6. Real time monitoring of strain change in TBM main load-bearing substructure during service tunneling. Finally, the feasibility of the method is verified by comparing the field measured data with the advanced prediction data.

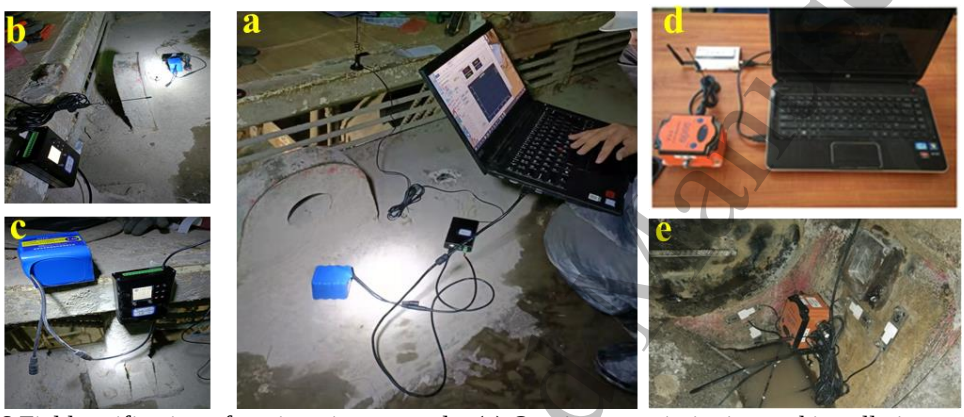


Figure 6 Field verification of engineering example. (a) On-site commissioning and installation stage, (b) Acquisition and monitoring wireless module, (c) In-situ monitoring system, (d) Remote online monitoring by the host computer, (e) The key structural position of the installation equipment of the strain monitoring package node.

3.1 Introduction to the research dataset and division of the training set

The dataset used in this study is from the actual working strain data of the key structure of the TBM cutter in the EH Diversion Water Supply Project, which was sampled 200 times per second, yielding a total of about 40,000 strain data. Therefore, due to the dataset characteristics, this study relies on the historical strain data to predict the future load data changes is a univariate prediction problem. In the strain time series dataset, the training set, validation set and test set are constructed in the ratio of 7:2:1. The training set is used to train the model for continuous optimization, the validation set can evaluate the prediction accuracy and robustness of the trained model and select the optimal model hyperparameters, and finally, it is used to evaluate the performance and generalization ability of the final model by comparing it with the real data in the test set, and the division and correlation of the three are shown in Figure 7.

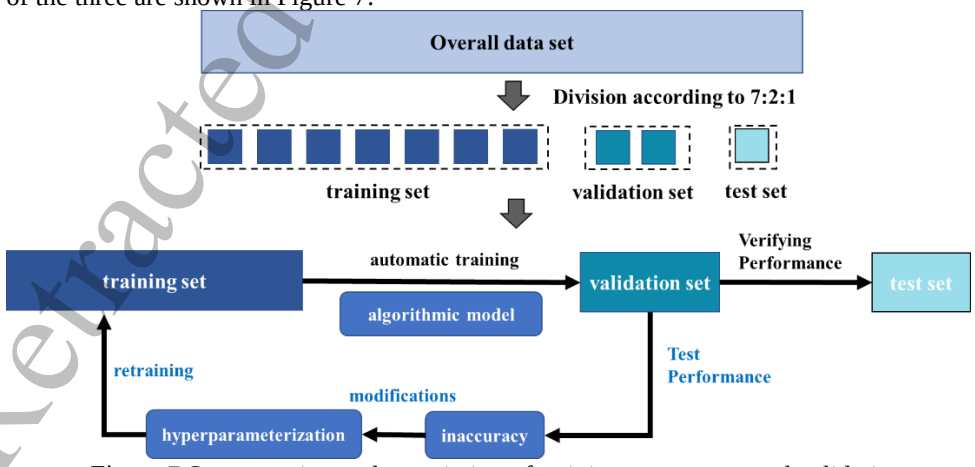


Figure 7 Segmentation and association of training set, test set and validation set

In order to effectively compare and validate the effect between different fatigue load prediction models, the root mean square error RMSE and the mean absolute percentage error MAPE are selected in this experiment to

evaluate the gap between the predicted and real values. The calculations of the two metrics are shown in Eq. 10 and 11, respectively.

$$RMSE(y,\hat{y}) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2)} \tag{10}$$

$$MAPE(y,\hat{y}) = \frac{100\%}{m}\sum_{i=1}^{m}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{11}$$

where m is the number of samples, y_i is the true value, $\hat{y}_i$ is the predicted value of the model, and $\bar{y}$ is the mean value of y.

3.2 Data Preprocessing

3.1.1 Comparative analysis of data filtering methods

Median filtering is a nonlinear digital filter technology, commonly used to remove noise in images or other signals, has a better filtering effect on noise, while being able to protect the detailed characteristics of the signal. The specific algorithm is: through a certain length of the sliding window on the time series processing, the initial input and the end part of the length of less than one-half of the window length (if odd, then round down) when the data remain unchanged, the middle part of the data, that is, to select the median of the data in the window as the corresponding position of the output. The median can be filtered several times in the program to produce a new data set. Assuming that the length of the input sequence is 20, the number of repetitions is 1, and the size of the sliding window is 5, then the process is as follows: x_1,x_2,x_{19},x_{20} remain unchanged, $x_3 = \{x_1,x_2,x_3,x_4,x_5\}$ median, $x_4 = \{x_2,x_3,x_4,x_5,x_6\}$ median, and so on, until the last window $x_{18} = \{x_{16},x_{17},x_{18},x_{19},x_{20}\}$ median. The model is optimized to get the best filtering scheme by adjusting the variables: the window length and the number of repetitions in the program.

The mean filtering process is similar to the median filtering, which only needs to transform the window to take the median value to take the mean value in the window, so the mean filtering is easy to be swayed by the extreme points or peaks in the window, which can not retain the edge information of the sequence well. The principle of wavelet filtering is to select the wavelet basis function, convert the original signal from the time domain to the wavelet domain, process the wavelet coefficients, and then convert the processed wavelet coefficients from the wavelet domain back to the time domain to get the denoised signal.

In order to facilitate the comparison of the effects of different filtering methods on the prediction effect of the model, the hyperparameters such as convolution kernel size, number of output channels and number of nodes in the hidden layer in the model were kept unchanged when replacing different filtering methods. After adjusting the values of optimization variables for each scheme, the final prediction effect is derived as shown in Table 2.

Table 2 Comparison of indicators of prediction results of different filtering methods

Filtering Methods	RMSE	MAPE
No filtering	66.2636	27.5486
Wavelet Filter	47.0003	22.4701
Mean Filter	37.1767	17.0486
Median filter	34.7907	16.3845

From the data in the table, it can be seen that the model with each filtering method added is better than the no-filtering case, so it is necessary to filter the original load data. In addition, among the three filtering methods, median filtering can make the best predictive model, increasing the RMSE index by 47% and the MAPE index by 41%. Therefore, median filtering was chosen as the data filtering method in this study. By adjusting the median filtering variables to optimize the results, the median filtering with a window length of 7 and a repetition number of 1 was finally applied to the data set, and the filtering effect is shown in Figure 8.

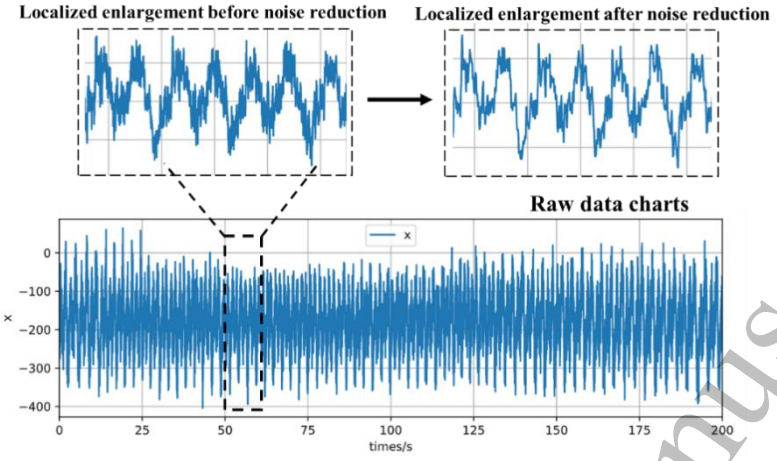


Figure 8 Comparison between median filtered data and original data

3.1.2 Data standardization

In the process of training neural networks, in order to eliminate the problem of difficult or impossible convergence of the model caused by the influence of different scales in the input variables as well as data features that are too large or too small, it is usually necessary to standardize the original data. Thus, a new data set with a mean of 0 and a standard deviation of 1 that obeys a standard normal distribution is obtained, and the specific transformation formula is shown in Equation 12.

$$x' = \frac{x - \mu}{\sigma} \tag{12}$$

where μ is the mean of all sample data and σ is the standard deviation of all sample data.

In this study, the data are standardized by configuring the standardization method Standard Scaler function, and in order to prevent the information of the test set from leaking to the training set, which affects the model training, the data are standardized after the training set and the test set are divided and then the data are standardized in the training set and the test set respectively. Among them, the data images after standardization of the training set and test set are shown in Figure 9.

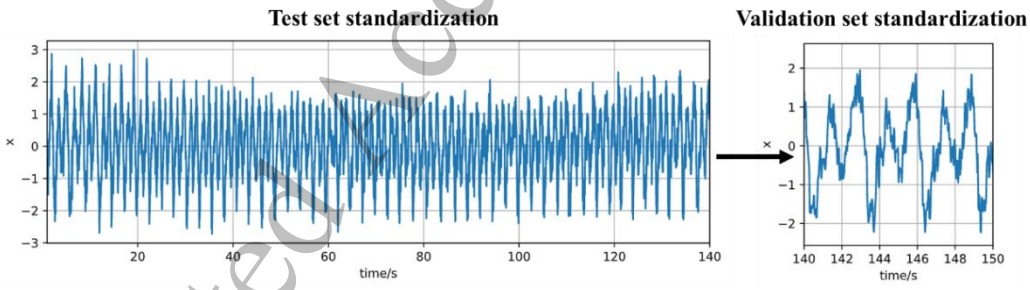


Figure 9 Normalized image of test set and validation set

3.3 Transformation of time series data into supervised learning

This study adopts multi-step prediction, i.e., predicting the future n values (also called n-step prediction) through an input window (h-length time series data) at each - prediction time. Here contains two hyperparameters: input window length h and multi-step prediction step *n*. After determining the input window length and multi-step prediction step, then use the sliding window to complete the construction of time series prediction samples of the specified length. The specific process is shown in Figure 10, assuming that the input window length *h* is 6, the multi-step prediction step n is 3, and the overall sequence length l is 21. Firstly, we construct the first training sample $\{X_1:(x_1, x_2,..., x_6); Y_1:(x_7, x_8, x_9)\}$; and then slide the input window backward by 1 time length to construct the second training sample $\{X_2:(x_2, x_3,..., x_7); Y_2:(x_8, x_9, x_{10})\}$; and then slide back the input window by 1 time length to construct the second training sample $\{X_{13}:(x_{13}, x_{14},..., x_{18}); Y_{13}:(x_{19}, x_{20}, x_{21})\}$. Based on the above analysis, it can be seen that: the number of final training set building block samples in the example = l-h-n+1 = 13.

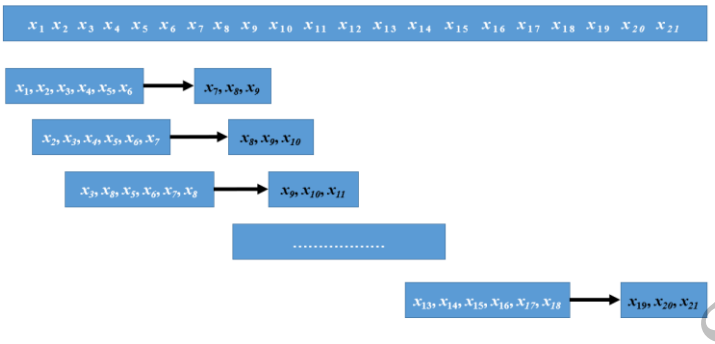


Figure 10 Schematic diagram of sliding window transition supervised learning

If all samples are trained one by one and the model is updated, the gradient descent uses the full batch of samples. However, this not only takes a longer time and reduces the efficiency of the model, but also corrects the direction of each correction in the direction of the gradient of the respective samples, which may make it difficult for the model to converge in the end. In order to improve the efficiency of model training and learning, the samples were divided into equal subsets in the experiments. The batch partitioning of the dataset is implemented through DataLoader, which combines the dataset and sampler and provides single- or multi-threaded iterable objects on the dataset. Through tuning optimization, in this study, the epoch is set to 40, the batch size is 64, shuffle=True, and the length of one prediction is 5. In the final prediction using the trained completed model, the prediction step size and the training sample construction step size should be consistent. However, no repeated prediction is made for a certain moment of data, so each input window should be slid back n steps in length of time to ensure that there is no missing and no repeated prediction data. The specific process is shown in Figure 11 (the dashed line boxed part indicates the data to be predicted), assuming that the known historical length of time and the length of the input window *h* is 6, the multi-step prediction step *n* is 3, and the length of the predicted data is 15. Each time the input window should be slid backward by 3 lengths of time, and the predicted value of the previous model step is inputted into the model as the real value, and then cycle the process to complete the prediction of a specified length of the task.

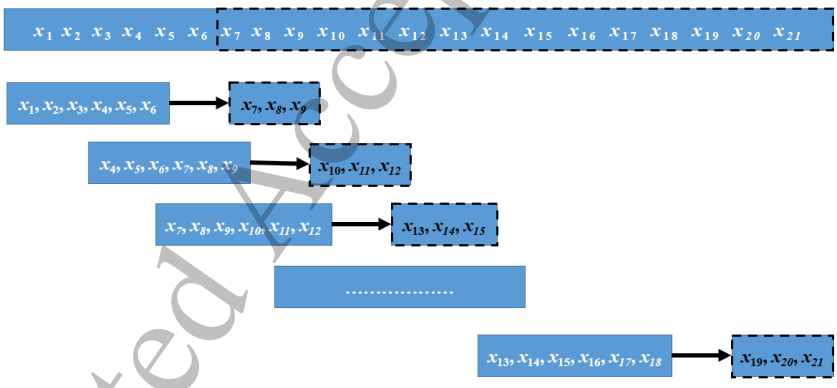


Figure 11 Schematic diagram of the sliding window prediction process

4 Results and Discussion

After processing the raw data and turning it into supervised learning, it can be sent to the model training to observe the model prediction results. The CNN-LSTM network prediction model with the addition of the attention mechanism includes the design of the overall structure of the model, the multi-condition identification module, the choice of the loss function, and the model experimental configuration. Ablation experiments were designed for comparison, and the final prediction results were analyzed by model hyperparameter adjustment, which proved the effectiveness and accuracy of the overall structure of the prediction model designed in this study.

4.1 Structure of time series prediction model

4.1.1 Attention Mechanism in CNN (ECA-Net)

ECA-Net (Efficient Channel Attention for Deep Convolutional Neural Networks) is an efficient channel attention module for deep CNNs that, after channel-by-channel global average pooling without dimensionality

reduction, considers each channel and its k-nearest neighbors to capturing local cross-channel interactions. Where k is determined by an adaptive method and the adaptive function is shown in Eq. 13. The coverage of cross-channel interactions (i.e., kernel size k) is seen to be proportional to the channel dimension through Eq.

$$k = \left| \frac{\log_2(c)}{\gamma} + \frac{b}{\gamma} \right| \tag{13}$$

where $\gamma = 2$, $b = 1$, and c is the number of channels. The overall computational process is as follows: the input feature map is subjected to global average pooling, and according to the number of channels in the feature map, the adaptive 1D convolution kernel size k is obtained. the k size convolution kernel is used for 1D convolution, and then the Sigmoid function is carried out to obtain the weight of each channel for the feature map, and the normalized weight and the original input feature map are multiplied channel by channel to generate the weighted feature map. channel by channel to generate the weighted feature map. The specific process is shown in Figure 12.

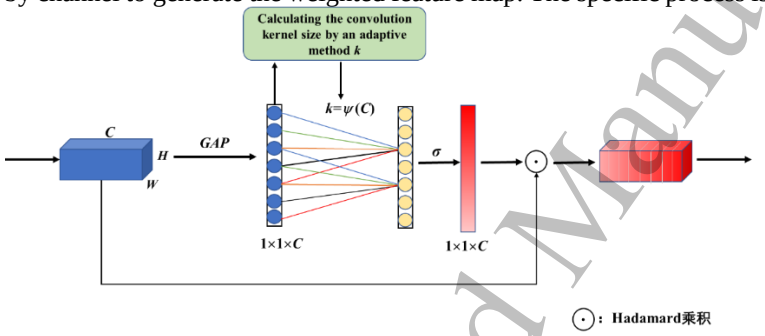


Figure 12 Calculation process diagram of ECA module

4.1.2 Overall network architecture design and analysis

ECA-Net is the attention mechanism of CNN, which can obtain the degree of importance of each channel in the feature map after CNN processing and give each channel a different attention score (weight value), so that the model pays more attention to the channels that have a greater impact on the results and suppresses the feature channels that do not have much impact on the results. LSTM is a kind of temporal recurrent neural network. It is mainly used to solve the long-term dependency problem that exists in RNN (Recurrent Neural Network). Relying on its gating unit and memory structure, it can effectively handle long time series data of a certain length. However, when faced with excessively long sequences, instability and gradient vanishing may occur when a single LSTM is trained, so it cannot capture very long-term interdependence.

Therefore, this study synthesizes the three parts of CNN, LSTM and ECA-Net, and the pre-processed signals are passed into the model, which first learns the high-dimensional abstract representation of the original load data through two-layer CNN, outputs the optimal representation of the relatively stable time-series signals, and pays more attention to the important feature extraction channels by giving different attention scores to each channel through ECA-Net. Then it is conveyed to the LSTM network to process the short-sequence high-dimensional features, and finally outputs the final timing prediction data through the fully connected layer. Its overall structure is schematically shown in Figure 13.

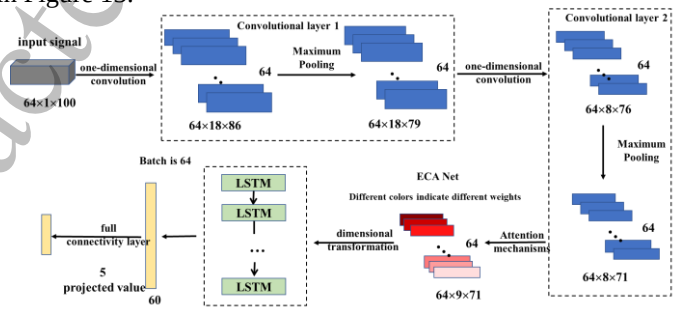


Figure 13 Schematic diagram of the overall structure of the model

4.1.3 Model Multi-Condition Recognition Module

Since the working conditions of key components of major equipment are more variable during service, if the component load changes from one relatively stable state to another, as shown in Figure 14. At this time, it is obviously inappropriate to utilize all the load data in the past time as the training dataset for prediction. This not

only makes the amount of learning data too much, the model training time is lengthened, and the prediction results of the new working conditions will have a large deviation. Therefore, the prediction model in this study incorporates a multi-working condition recognition module.

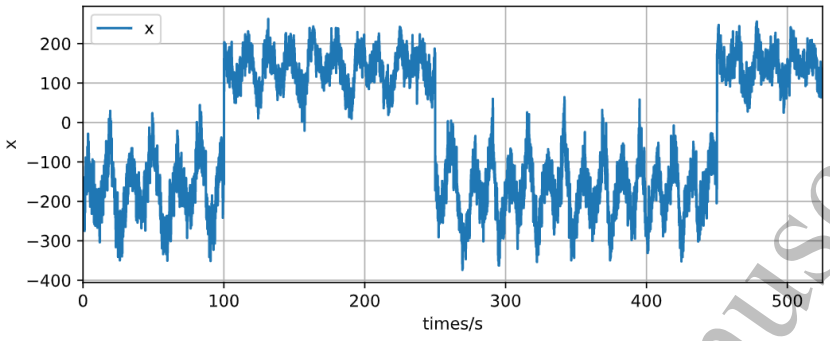


Figure 14 Schematic diagram of load variation in multiple working conditions

The working principle of this module is to set the detection interval and the predetermined value in advance according to the actual working requirements, in which the detection interval determines how long the average value is calculated every time, and the predetermined value determines how much the change exceeds the degree to be regarded as entering a new working condition, and the process is as shown in Figure 15. For the input data, the average value of the data is calculated for each period of time according to the detection interval, and then it is judged whether the change of the average value within the time interval exceeds the set predetermined value. If it does not exceed the set predetermined value, it is normally input into the model for training and prediction. If the change in the mean value exceeds the set predetermined value, a prompt is issued and the dataset range is re-selected, and the load data prior to the change is discarded before it is re-input into the model for training and prediction.

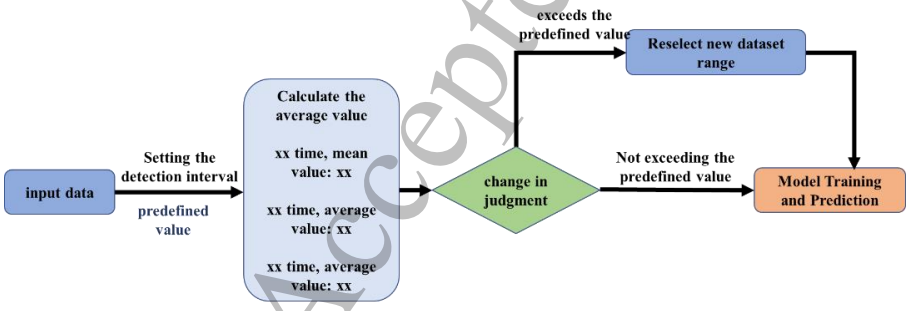


Figure 15 Schematic diagram of the principle of model multi-condition recognition module

By adding the multi-case identification module, the prediction model can deal with a wider range of datasets, more diverse application scenarios, and more in line with the actual service process of the mechanical equipment load variable situation, so as to be better applied to the actual fatigue load prediction of major equipment, speed up the prediction speed, and improve the prediction accuracy.

4.2 Introduction to loss function analysis

Supervised learning in deep learning is essentially the process of trying to learn the $x \rightarrow y$ mapping given a series of training samples (x_i,y_i) so that given an x (even if it is not in the training samples), the output ŷ is as close as possible to the true y. The Loss Function is a key part of this process and is used to measure the difference between the model's output $\hat{y}$ and the true y, giving direction to the model optimization. and the real y, to give direction to the optimization of the model. This subsection introduces the common loss functions and the Smooth L1 loss function chosen for this study.

4.2.1 Common MSE, MAE loss functions

Mean Square Error (MSE) is the most commonly used error in the regression loss function, which is the mean of the sum of the squares of the differences between the predicted value $f(x)$ and the target value y. Its mathematical calculation is shown in Equation 14.

$$MSE = \frac{\sum_{i=1}^{n}(f(x)-y)^2}{n} \tag{14}$$

The curve distribution of the mean square error value is shown in Figure 16, which shows that the MSE loss function increases gradually as the difference between the predicted value and the target value increases. Therefore, the MSE loss function is more sensitive to outliers and is more affected by them. And its advantage is that the curve is continuous, smooth, and conductible everywhere, which facilitates the use of gradient descent algorithm and helps the model to converge.

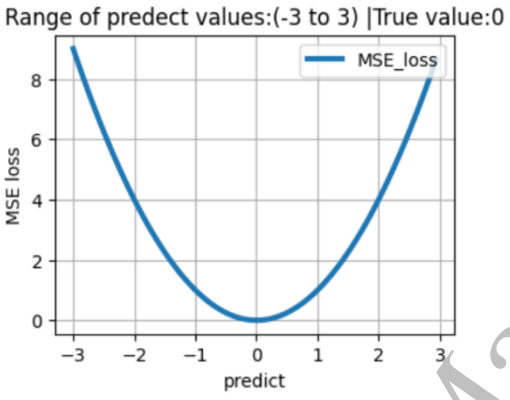


Figure 16 Plot of MSE losses vs. predicted values

The mean absolute error (MAE) is the average of the sum of the absolute values of the difference between the target value and the predicted value, and is another commonly used regression loss function, the mathematical calculation of which is shown in Equation 15.

$$MAE = \frac{\sum_{i=1}^{n}|f(x) - y|}{n} \tag{15}$$

The curve distribution of the mean absolute error value is shown in Figure 17, which shows that the curve is continuous and has a stable gradient for any size of difference, but is not derivable at f(x)=y. And because of its constant gradient, it is unfavorable for the convergence of the function and the learning of the model when confronted with an environment with small differences. Obviously, the advantage is that the use of the MAE loss function can greatly reduce the influence of outliers on the model.

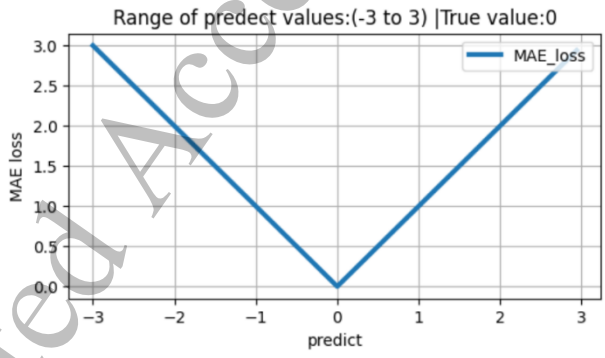


Figure 17 Plot of MAE losses vs. predicted values

4.2.2 L1-paradigm and L2-paradigm loss function

The L1-paradigm loss function, also known as the Least Absolute Error (LAE), and the L2-paradigm loss function, also known as the Least Squared Error (LSE), are computed as shown in Equations 16 and 17.

$$L1 = \sum_{i=1}^{n}|f(x) - y| \tag{16}$$

$$L2 = \sum_{i=1}^{n}(f(x) - y)^2 \tag{17}$$

It can be found through the formula that the L1-paradigm and L2-paradigm loss functions are only different from the MAE and MSE loss functions in the previous section by dividing by 1/n, so they have the same advantages and disadvantages.

4.2.3 Smooth L1 Loss Function

Smooth L1 loss function (also known as Huber loss function) is the L1 loss after smoothing. The disadvantage of L1 loss is that it has non-leadable points (folds) and the gradient is the same for any difference, which is not

conducive to the convergence of the function in the case of low differences. Smooth L1 optimizes the L1 loss function as a segmented function, whose algorithm is shown in Eq. 18 and 19. This loss function can limit the gradient in two ways: when the difference between the predicted value and the true value is large, the gradient value will not be too large, which reduces the impact of outliers on the model; when the difference between the predicted value and the true value is small, the gradient value is small enough to facilitate convergence and model training.

$$loss(x,y) = \frac{1}{n}\sum_{i} z_i \tag{18}$$

$$z_i = \begin{cases} \frac{1}{2\beta}(x_i - y_i)^2, & |x_i - y_i| < \beta \\ |x_i - y_i| - \frac{1}{2}\beta, & otherwise \end{cases} \tag{19}$$

Where x_i is the predicted value, y_i is the true value, and β is the threshold that specifies the change of this loss between L1~ L2, which defaults to 1.0 in the program. setting the true value to 0, the predicted value from -3 to 3, and β to 1, results in the Smooth L1 and L1 and L2 loss functions shown in Figure 18. From the figure, it can be seen that the Smooth L1 loss function is a segmented function, which is similar to the L2 loss in the [-1,1] interval, which solves the folding point of L1 as well as the problem of difficult convergence, and is basically the same as the L1 loss outside of the [-1,1] interval, which reduces the impact of outliers on the model.

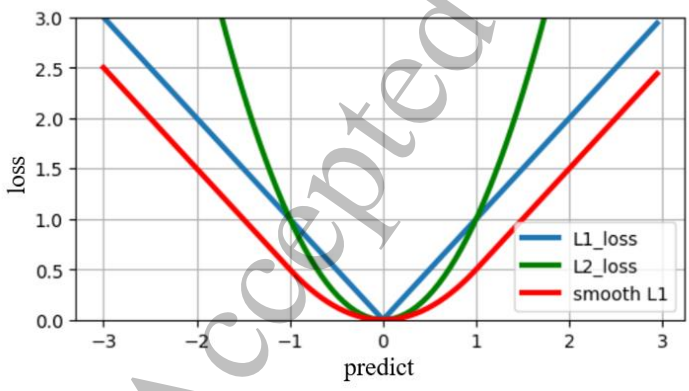


Figure 18 Comparison of Smooth L1 and L1 and L2 loss functions

In this study, Smooth L1 was finally chosen as the model loss function by applying the nn.SmoothL1Loss function in pytorch and setting β as 1.

4.3 Predictive Modeling Experimental Configuration

The computer configuration used in this study is: processor Intel(R) Core(TM) i7-9750H CPU with NVIDIA GeForce GTX 1650 accelerated training, the specific environment configuration shown in Table 3.

Table 3 Experimental hardware configuration and software environment

Configuration Environment	Version
CPU	Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz
GPU	NVIDIA GeForce GTX 1650
Memory	8GB
Operating Systems	Windows 10 64
Programming Languages	Python 3.9.12
Software packages	Pytorch 1.13.0，numpy 1.21.5

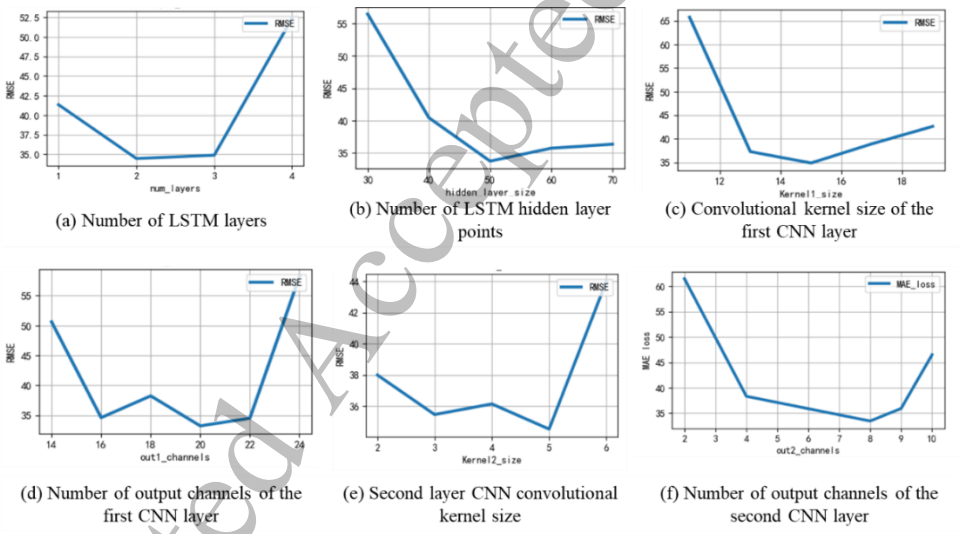
Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems. Received July 18, 2023;
Accepted manuscript posted June 06, 2024. doi:10.1115/1.4065819

Before the model starts training, median filtering with a window of 7 and a number of times of 1 is performed on the dataset, and then the filtered dataset is constructed into training, validation, and test sets in the ratio of 7:2:1, and data standardization is performed on the training and validation sets, respectively. Finally, the standardized data are used to create labels using the sliding window method, which are divided into batches and then put into the model training. In the experiments, the batch size is set to 64, the training period is 40, the learning rate is 2e-4, and the model is added with a Dropout layer to prevent overfitting, and the Dropout rate is set to 0.2. In addition, the model is optimized by using the Adam optimizer, which is a highly efficient computational tool that calculates different adaptive learning rates for different parameters, which can better optimize the model. optimize the model. At the same time, in order to better optimize the model, lr_scheduler, a class for learning rate adjustment, is referenced in the torch.optim package of Pytorch, and the parameters are set as follows: the learning rate is updated every 15 epochs, and the multiplication factor of each update is 0.1, i.e., the learning rate is adjusted to be 0.1 times of the original value for every 15 cycles.

4.4 Model Predictive Effectiveness Reconciliation and Analysis

4.4.1 Hyperparameter tuning and analysis

As can be seen from the previous overall model structure setup, the hyperparameters of the model are mainly concentrated in the CNN and LSTM layers, in which each layer of the CNN part of the convolution operation contains three hyperparameters: the size of the convolution kernel, the number of convolution kernels (i.e., the number of output channels), and the size of the pooling kernel of pooling layer. (multi-layer LSTM, i.e., the output of the previous LSTM layer is used as the input of the subsequent LSTM layer). In order to determine the optimal combination of hyperparameters and to consider the efficiency of the tuning, one hyperparameter is tuned at a time by fixing the other hyperparameters and setting the set of values to observe the loss in the test set. Some of the hyperparameter adjustments and the RMSE evaluation results of their corresponding models are shown in Figure 19.



(a) Number of LSTM layers
(b) Number of LSTM hidden layer points
(c) Convolutional kernel size of the first CNN layer
(d) Number of output channels of the first CNN layer
(e) Second layer CNN convolutional kernel size
(f) Number of output channels of the second CNN layer

Figure 19 Plot of hyperparameter loss results in the regulation section

From the above figure, it can be seen that with the increase of hyperparameters, the prediction results of the model show a tendency of getting better first and then gradually getting worse. This is because as, for example, the input step size, the number of hidden layer nodes, and the number of convolutional kernels increase, the amount of time-series data learned by the model in a single session gradually increases, and the fitting ability is enhanced. When a certain range is reached, overfitting phenomenon easily occurs due to the model being more complex and difficult to train, which even leads to problems such as unstable training process or failure to converge, etc. Figure 20 shows the prediction result plots after changing the size of the first CNN convolutional kernel to 11,13,15, respectively. The prediction result graph can be more intuitively seen that when the convolutional kernel size is small, the model's fitting learning ability is insufficient, and it cannot better predict the future load data trend. When the convolution kernel size is large, the model is partially overfitted, and it is also difficult to predict future load changes well.

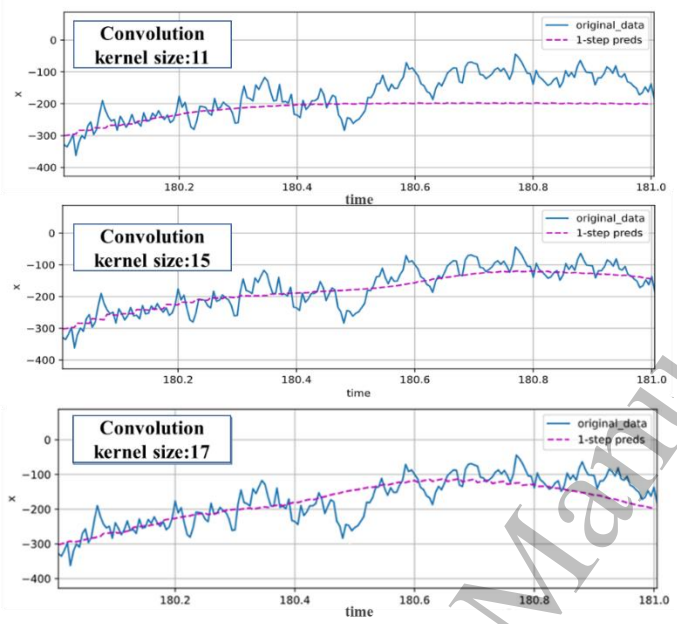


Figure 20 Comparison of prediction results after changing the size of the convolutional kernel of the first CNN layer

Therefore, by comparing the experimental results, the optimal structure and hyperparameter configuration of the model were finalized as shown in Table 4.

Table 4 Table of optimal hyperparameter configurations for the model

model	Hyperparameter	Numerical
First Convolutional Layer	Convolution kernel size	15
	Number of output channels	18
	Pooling layer pooling kernel size	8
Second Convolutional Layer	Convolution Kernel Size	4
	Number of Output Channels	8
	Pooling layer pooling kernel size	6
LSTM layer	Input step size	100
	Number of hidden layer nodes	64
	Number of layers	2

4.4.2 Analysis of model predictions

After setting the optimal hyperparameters of the model, the model is trained according to the model configuration. The loss function optimization process during the training set and validation set training is shown in Figure 20, through which it can be seen that the loss function basically no longer changes after 20 times of training and the model converges. The comparison of the final prediction results with the real data in the test set is shown in Figure 21, which shows that the noise is difficult to predict in the actual working conditions, but the prediction results can fit the data trend better. By analyzing the final data, among the 200 data lengths predicted, there are 89 data with prediction accuracy of 90%, accounting for 44.5%. there are 51 data with prediction accuracy of 80% to 90%, accounting for 25.5%. the RMSE of the prediction results is 33.2626, and the MAPE is 15.37677.

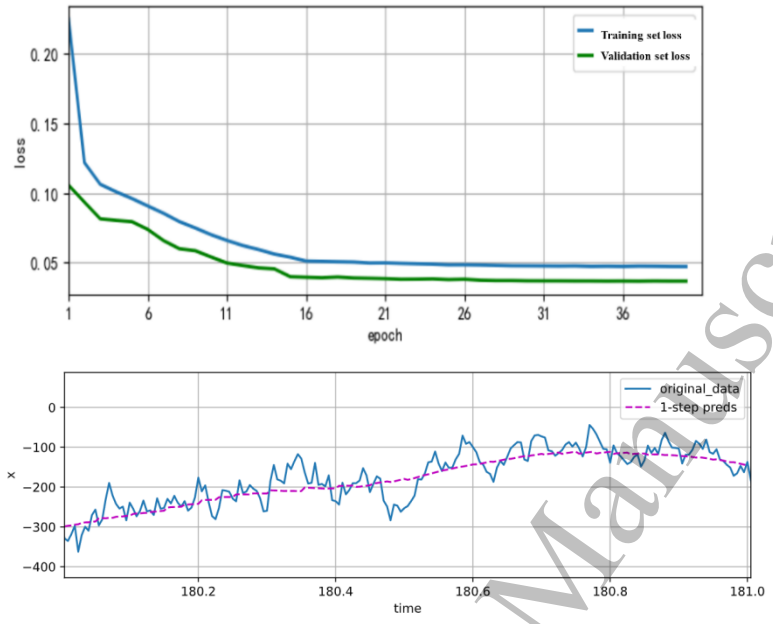


Figure 21 Plot of loss function changes during model optimization and model prediction results

4.4.3 Comparative analysis of ablation experiments

This model consists of three parts: the CNN, ECA-Net, and LSTM. they are respectively responsible for learning the high-dimensional abstract representation of the original load data and outputting the optimal representation of the relatively stable time-series signals; obtaining the importance degree of the channels in the feature map and assigning different attention scores; and processing the high-dimensional features of the time-series and outputting the final prediction result. In order to verify the effectiveness and accuracy of the overall model, a module ablation comparison experiment is conducted on the attention-based CNN-LSTM network to compare the prediction results with the three models of LSTM, CNN, and LSTM-CNN. The comparison results are shown in Table 5.

Table 5 Comparison of prediction results of different models

Model	RMSE	MAPE
LSTM	35.2637	17.0806
CNN	37.3577	17.8413
Attention-CNN	36.6791	16.9127
CNN-LSTM	34.9746	16.6774
Attention-CNN-LSTM	33.2626	15.3768

The comparison of the prediction results of different models shows that the CNN-LSTM network with the addition of the attention mechanism has the best prediction results, and the worst prediction results are obtained when only CNN is used for prediction. This indicates that compared to LSTM networks, CNN cannot handle time series data well and is difficult to capture the correlation between long series data directly. Meanwhile, from the tabular data, it can also be seen that the CNN-LSTM network prediction is better than the CNN or LSTM model alone, which means that the use of CNN to extract the high-dimensional features abstracted from the original data in conjunction with the advantages of LSTM in processing long-series data can improve the model's learning ability and prediction accuracy. However, due to its lack of attention mechanism, when the number of output channels of CNN increases, the processing effect of the model tends to decrease instead, and the prediction accuracy is difficult to rise further. Therefore, through the comparison of different experimental models, it proves the effectiveness of the overall structure of the CNN+LSTM network prediction model designed in this study by adding the attention mechanism, so that the overall structure of the three parts of CNN, ECA-Net, and LSTM improves the accuracy of prediction.

5 Conclusion

As an important embodiment of scientific and technological level innovation, the development of major equipment is related to China's comprehensive strength and international status. In this paper, the study uses the actual working strain data of the key structure of TBM cutter as the experimental data set, and carries out the load

prediction based on the powerful nonlinear fitting ability and multimodal fusion ability of deep learning, and the main completed work and conclusions are as follows.

(1) Due to more disturbing factors in the service process of major equipment, there are more noise data and other problems in the actual load data, so the original data need to be preprocessed. In this paper, by comparing the prediction results of different filtering methods, the median filter with a window of 7 and a repetition number of 1 is finally selected, which improves the RMSE index by 47% and the MAPE index by 41% compared with the unfiltered model.

(2) In this paper, a CNN-LSTM network prediction model incorporating an attention mechanism is proposed and validated. After analyzing and optimizing the hyperparameters, the best hyperparameters are determined so that the model has the best fitting ability before training, and the future 200 data points are predicted, and the prediction accuracy reaches 90% with 89 data, accounting for 44.5%, and the final prediction result has an RMSE of 33.2626 and a MAPE of 15.37677.

(3) In order to prove the accuracy of each component structure of the prediction model designed in this paper, Smooth L1 is finally chosen as the loss function of the model, and the RMSE of predicting the future 200 lengths of time-series data is 33.2626, and the MAPE is 15.37677. ablation experiments are designed to compare the comparisons, and the comparison of the evaluation indexes concludes that the model proposed in this paper has the best overall prediction performance, and the overall prediction performance of the model proposed by analyzing the LSTM, CNN, CNN-LSTM, and the prediction result metrics of the model, the effectiveness of the overall structure of the prediction model designed in this study and the necessity of the three components of CNN, ECA-Net, and LSTM are proved.

Highlights

(1) Preprocessing load data by comparing different filtering methods.

(2) A CNN-LSTM network prediction model incorporating an attention mechanism is proposed.

(3) An ablation experiment is designed to predict the future 200 lengths of time series data.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

National Natural Science Foundation of China (No. 52275236) ； "Unveiling the List and Leading" Major special science and technology project of Liaoning Province (No. 2022JH1/10400031)

Data availability statement

Research data are not shared.

References

[1] Rebecca S, Esther P, Eduardo B, et al. Eduardo O. TSPred: A framework for nonstationary time series prediction[J]. Neurocomputing,2022,467.

[2] H.T. Liang, S. Liu, J.W. Du, et al. A review of research on deep learning applied to timing prediction[J]. Computer Science and Exploration:1-21.

[3] Dong Y X, Xiao L, Wang J, et al. A time series attention mechanism-based model for tourism demand forecasting[J]. Information Sciences,2023,628.

[4] Xue Yanjie. Time series prediction of deep foundation pit deformation in soil-rock composite stratum based on machine learning algorithm[J]. Modern Tunneling Technology,2022,59(S2):77-85.

[5] Klepsch J, Klüppelberg C, Wei T. Prediction of functional ARMA processes with an application to traffic data[J]. Econometrics and Statistics,2017,1.

[6] Bezerra S M C, Marcio D C M, Didier L I, et al. Remaining Useful Life Estimation by Empirical Mode Decomposition and Support Vector Machine[J]. IEEE Latin America Transactions,2016,14(11).

[7] Song Xueguan, Lai Xiaonan, He Xiwang, et al. Key technology of digital twin for major equipment formability integration[J]. Journal of Mechanical Engineering,2022,58(10):298-325.

[8] Que Zijun. Fault prediction of thermal power generating units based on temporal feature machine lea rning [D]. Zhejiang University,2022.

[9] Voyant C, Muselli M, Paoli C, et al. Numerical weather prediction (NWP) and hybrid ARMA/ANN model to predict global radiation[J]. Energy,2012,39(1).

[10] Klepsch J, Klüppelberg C, Wei T. Prediction of functional ARMA processes with an application to traffic data[J]. Econometrics and Statistics,2017,1.

[11] WANG Yao,ZHAO Zhirong. Statistical analysis of SSE monthly average closing price trend - based on ARMA model[J]. Journal of Taiyuan Normal College(Natural Science Edition),2019,18(04):40-42+64.

[12] Maria S, Pawel P. Temporal Pattern Attention for Multivariate Time Series of Tennis Strokes Classification[J]. Sensors,2023,23(5).
[13] Gao Yang, Zhang Biling, Mao Jingli, et al. Adaptive photovoltaic ultra-short-term output prediction model based on machine learning[J]. Grid Technology,2015,39(02):307-311.
[14] Xue Yanjie. Time series prediction of deep foundation pit deformation in soil-rock composite stratum based on machine learning algorithm[J]. Modern Tunneling Technology,2022,59(S2):77-85.
[15] Lee D, Lee J, Kim Y, et al. Thermo mechanical fatigue life prediction of Ni-based superalloy IN738LC[J]. International Journal of Precision Engineering and Manufacturing,2017,18(4).
[16] Liu Y, Liu Z Z, Zuo H F, et al. A DLSTM-Network-Based Approach for Mechanical Remaining Useful Life Prediction[J]. Sensors,2022,22(15).
[17] Ding Ming, Wang Lei, BI Rui. Short-term prediction model of photovoltaic power generation system output power based on improved BP neural network[J]. Power System Protection and Control,2012,40(11):93-99+148.
[18] Hong W, Wang S P, Tomovic M, et al. A Novel Indicator for Mechanical Failure and Life Prediction Based on Debris Monitoring[J]. IEEE Transactions on Reliability,2017,66(1).
[19] X. Wang, J. Wu, C. Liu, et al. Fault time series prediction based on LSTM recurrent neural network[J]. Journal of Beijing University of Aeronautics and Astronautics,2018,44(04):772-784.
[20] Yang Qing, Wang Chenwei. Research on global stock index prediction based on deep learning LSTM neural network[J]. Statistical Research,2019,36(03):65-77.
[21] Jun H, Wen D Z. A deep learning model to effectively capture mutation information in multivariate time series prediction[J]. Knowledge-Based Systems,2020,203.
[22] Xie G, Shangguan A Q, Fei R, et al. Motion trajectory prediction based on a CNN-LSTM sequential model[J]. Science China Information Sciences,2020,63(11).
[23] Wang Q, Wu B, Zhu P, et al. ECA-Net: Efficient channel attention for deep convolutional neural networks[J]. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,2020.
[24] Du Shengdong, Li Tianrui,Yang Yan, et al. A traffic flow prediction model based on sequence-to-sequence spatio-temporal attention learning[J]. Computer Research and Development,2020,57(08):1715-1728.
[25] Gou, Chao, Zhou, et al. Driver attention prediction based on convolution and transformers[J]. The Journal of Supercomputing,2022.
[26] Yingying Wu, Lining Zhao, Zhixin Yuan, et al. A CNN-GRU ship traffic flow prediction model based on attention mechanism[J]. Journal of Dalian Maritime University,2023,49(01):75-84.