Nondestructive Testing and Evaluation

ISSN: 1058-9759 (Print) 1477-2671 (Online) Journal homepage: www.tandfonline.com/journals/gnte20

A sequence ensemble method based on Sparse Transformer with bidirectional gated recurrent convolution unit for remaining useful life prediction

Changbo He, Zehua Fan, Peng Chen, Wei Fan, Zhixiong Li & Hongkun Li

To cite this article: Changbo He, Zehua Fan, Peng Chen, Wei Fan, Zhixiong Li & Hongkun Li (08 Sep 2025): A sequence ensemble method based on Sparse Transformer with bidirectional gated recurrent convolution unit for remaining useful life prediction, Nondestructive Testing and Evaluation, DOI: 10.1080/10589759.2025.2558066

To link to this article: https://doi.org/10.1080/10589759.2025.2558066

Published online: 08 Sep 2025.

NONDESTRUCTIVE TESTING AND EVALUATION
https://doi.org/10.1080/10589759.2025.2558066

Taylor & Francis
Taylor & Francis Group

A sequence ensemble method based on Sparse Transformer with bidirectional gated recurrent convolution unit for remaining useful life prediction

Changbo He[a], Zehua Fan[a], Peng Chen[b], Wei Fan[c], Zhixiong Li[d] and Hongkun Li[e]

[a]School of Electrical Engineering and Automation, Anhui University, Hefei, China; [b]College of Engineering, Shantou University, Shantou, Guangdong, China; [c]School of Mechanical Engineering, Jiangsu University, Zhenjiang, China; [d]Faculty of Mechanical Engineering, Opole University of Technology, Opole, Poland; [e]School of Mechanical Engineering, Dalian University of Technology, Dalian, China

ABSTRACT

The remaining useful life prediction (RUL) of rotating machinery is crucial for Prognostic Health Management (PHM) tasks. Relying on deep learning methods, the RUL prediction task of mechanical facility has made good progress. However, existing methods often do not consider mechanical degradation rules and cannot effectively capture short-term and long-term feature dependencies. To solve such problems and achieve a better RUL prediction effect, this paper designs a sequence ensemble model. The model synergistically integrates two key components: a bidirectional gated recurrent convolution unit (Bi-GRCU) and Sparse Transformer. This integration enables comprehensive dependency learning across different time scales. Specifically, the Bi-GRCU combines the strengths of recurrent and convolutional networks to effectively capture short-term dependencies. Meanwhile, the Sparse Transformer employs long-range locality sparse attention (LRLS-Attention), which can efficiently identify long-term dependencies by filtering redundant computations and reducing dot-product complexity. By sequentially integrating these two components, the proposed model achieves a balanced representation of both local and global temporal patterns, significantly improving prediction accuracy and robustness. The related experiments are conducted by applying the CMAPSS aeroengine dataset and bearing wear PHM2010 dataset, and the results demonstrate that the predictive ability of proposed model is superior to other existing methods.

ARTICLE HISTORY
Received 6 June 2025
Accepted 2 September 2025

KEYWORDS
Rotating machinery; PHM; remaining useful life; Bi-GRCU; Sparse Transformer

1. Introduction

Prognostic Health Management (PHM) technology is vital in industrial and manufacturing sectors for accident prevention and system reliability [1,2]. A key component of PHM is the accurate prediction of remaining useful life (RUL) of machinery. The accurate RUL prediction is crucial for minimising downtime, cutting maintenance costs and ensuring reliable operation, especially in industries where unexpected failures can lead to

CONTACT Changbo He changbh@ahu.edu.cn

significant financial and safety issues. Thus, it is necessary to research on advanced RUL prediction methods [3,4].

The prediction methods of RUL can be simply separated into two categories: which are traditional statistical techniques and advanced deep learning techniques [5]. The former requires an accurate mathematical model construction of the target system based on a large amount of historical information [6], while the latter requires the data preprocessing of the target task. Usually, a suitable neural network model will be constructed to train the preprocessed data, so that predicting the RUL is feasible.

Recently, a growing number of researchers have begun to use prediction methods based on deep learning to perform prediction tasks. Unlike conventional machine learning techniques, deep learning-based prediction methods can help establish corresponding dependencies between historical data and RUL without prior knowledge. Besides, when facing complex data, deep learning-based methods can show excellent computing efficiency and accurate prediction results [7]. For example, Li et al. designed an approach for predicting RUL utilising deep Convolutional Neural Network (CNN) combined with fully connected layers, which effectively solved the problem of failing to extract local features in complex data [8]. Considering that working condition degradation data is prone to be missing, Huang et al. designed an innovative Bidirectional Long- and Short-Term Memory (Bi-LSTM) network that integrates the feature data of multiple sensors to effectively perform the RUL prediction task [9]. Zhang et al. dedicated to the integration of spatiotemporal features, integrating Bi-GRU with CNN to build the correspondence between RUL and features [10]. Additionally, Liu et al. improved the LSTM network to design a new network (ILSTMC) [11]. Shi et al. developed a new Dual-LSTM that integrates the construction of health index with change point detection, enabling precise estimation of RUL for complicated feature data [12].

With the emergence of Transformer model, numerous researchers have started exploring its application in RUL prediction. Unlike traditional RNNs and CNNs, the key benefit of the Transformer is its integration of self-attention mechanisms (SAM), which enables it to better obtain long-term dependencies when performing parallel computation, so it shows strong performance in RUL prediction. For instance, in order to identify the dependency between the characteristics of degraded data, Zhang et al. adopted the SAM and then fused the features by splicing them together [13]. Li et al. [14] applied domain-adaptive technology to Transformer model for RUL prediction, which improved the prediction accuracy. Liu et al. [15] innovated the Transformer network by combining Transformer network and channel attention CNN to achieve accurate RUL prediction.

Despite recent advances in deep learning have led to improved RUL prediction accuracy, they still have certain limitations.: (1) The CNN-based method needs to set a fixed size of convolution kernel when handling feature data, making the model difficult to capture distant features [13]. (2) The RNN-based method cannot create corresponding relationships between different feature data, which easily limits the ability of parallel learning [16,17]. (3) Transformer model strives to establish a global information dependency relationship but cannot effectively concentrate on local information [17]. In addition, when subtle changes occur in degraded data, it is insensitive to hidden degradation feature using Transformer model.

With the aim of overcoming these constraints, this article proposes a Sparse Transformer with Bi-GRCU network, which demonstrates exceptional prediction performance. The Bi-GRCU network is created by integrating the functional characteristics of CNN and RNN. This combination enables the model to effectively manage data dimensions and capture local dependencies at each time step. Such integration facilitates the extraction of critical features from time-series data, thereby enhancing prediction performance during model training. Additionally, the LRLS-Attention mechanism is proposed and incorporated into the Transformer's encoder structure to form the Sparse Transformer. This modification enables the model to assign different weights to various information positions within the input sequences based on their relationships with other positions. As a result, the Sparse Transformer can effectively establish correlations between different information positions and further enhance feature extraction from the output generated by the Bi-GRCU network. By sequentially integrating these two components, the proposed model achieves a balanced representation of both local and global temporal features, significantly improving prediction accuracy and robustness. The contributions of this work are summarised below:

(1) By combining bidirectional processing and gating mechanism, Bi-GRCU is proposed, which can effectively extract features, capture short-term dependencies and improve the local perception ability of data context.
(2) By optimising the sparse strategy, proposing LRLS-Attention and integrating it into Transformer to form Sparse Transformer. The Sparse Transformer can effectively identify long-term dependencies and filters out the main function of dot product calculation in time series prediction, greatly reducing computational complexity and storage requirements and improving prediction efficiency.
(3) Experimental verification was conducted using the CMAPSS aeroengine dataset and bearing wear dataset PHM2010. The results validated that the predictive ability of the Sparse Transformer with Bi-GRCU network. For instance, compared with other data-driven methods, the prediction performance using the CMAPSS dataset showed the best improvement of 18.18% and the PHM2010 dataset achieved the best improvement of 44.36%. The results indicated that the model was superior to other data-driven methods.

The structure of this paper is organised as follows: Section 2 explains the architecture of the Sparse Transformer with Bi-GRCU network and outlines the experimental process for the prediction task. Section 3 presents the details of the prediction task using the CMAPSS aeroengine dataset, along with a comparative analysis of the experimental results. Section 4 covers the prediction task using the PHM2010 dataset, including the related results. Finally, Section 5 presents a recap of the main findings of the paper.

2. Proposed model

2.1. Bidirectional Gated Recurrent Convolutional Unit (Bi-GRCU)

Bidirectional Gated Recurrent Convolutional Unit is a new network structure, which mainly consists of a linear layer, a scale perception convolution module (SPCM) and two

bidirectional GRUs. Each GRU unit contains a reset gate, a candidate hidden state and an update gate, which adjusts the flow and updating of information. Compared with traditional unidirectional recurrent networks, Bi-GRCU is able to simultaneously capture the before and after dependencies in the sequence. It can use both future and past feature information to predict the current output, thereby gaining a deeper understanding of the relationship of the sequence data. Figure 1 is the structure of GRU:

(1) Reset Gate:

$$r_t = sigmoid(W_{ir}x_t + W_{hr}h_{t-1} + b_r) \tag{1}$$

By learning the weights, Reset Gate determines how to mix new inputs with previous hidden states.

(2) Update Gate:

$$z_t = sigmoid(W_{iz}x_t + W_{hz}h_{t-1} + b_z) \tag{2}$$

Update Gate is used to control the integration of the current input information and the previous hidden state.

When Bi-GRCU extracts data features, the hidden state is updated using these gates. The hidden state equation is:

$$\tilde{h}_t = \tanh(W_{ir}x_t + W_{ih}(h_{t-1} \otimes r_t) + b_h) \tag{3}$$

Finally, z_t combine r_t and h_{t-1} to form Equation (4):

$$h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes \tilde{h}_t \tag{4}$$

$W_{iz} \in R^{m\times m}, W_{hz} \in R^{m\times m}, W_{ir} \in R^{m\times m}, W_{hr} \in R^{m\times m}$, which is a weight matrix. $b_r, b_z \in R^m$is a deviation, $\tilde{h}_t$ is a candidate hidden state, h_t is an updated hidden state, $\otimes$ representing element wise multiplication.

When using Bi-GRCU for feature extraction, the input sequence x_t first passes through the SPCM. Specially, the input data is first processed through a 1×1 convolutional layer with batch normalisation and ReLU activation function, then it enters another 1×1 convolutional layer followed by two 3×3 convolutional layers for feature

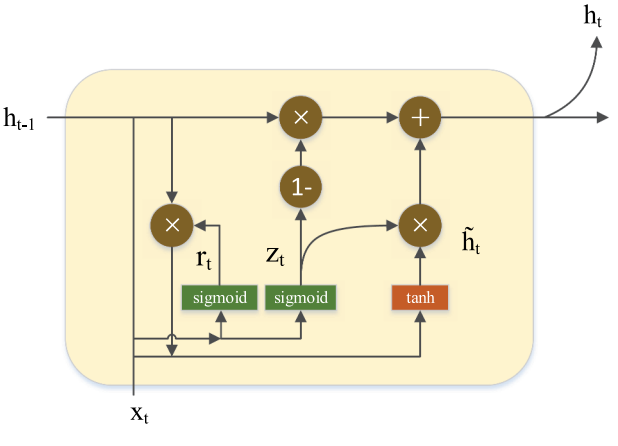


Figure 1. The overall structure of GRU.

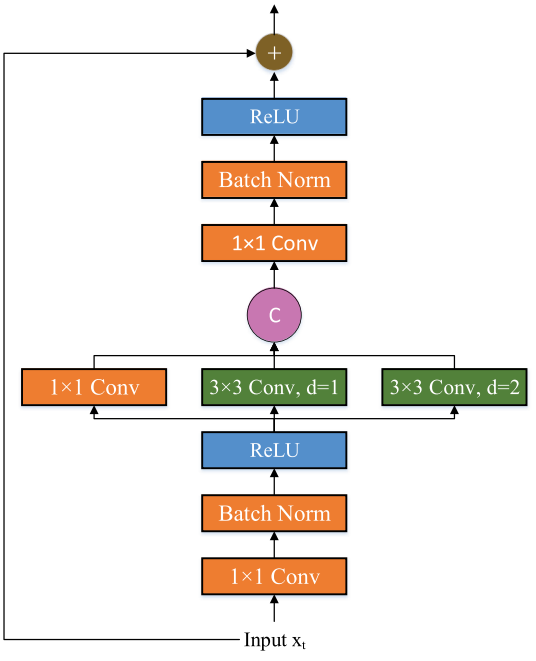


Figure 2. The structure of the SPCM.

extraction. Their outputs are concatenated and passed through a final 1×1 convolutional layer with batch normalisation and ReLU activation function. The result is then added to the input data via a residual connection. Figure 2 is the structure of the SPCM. *d* denotes the dilation size.

SPCM can not only extract multi-dimensional features but also assist GRU in capturing detailed short dependencies, thereby extracting local features from input sequences. The 1×1 convolutional kernel retains key features while reducing dimensionality and the 3×3 convolution layers focus on capturing local spatial information. By concatenating multi-layer outputs, this module effectively integrates local features across different scales and further optimises through another convolution layer. Additionally, the residual connection can prevent gradient vanishing, enabling GRU to process these features more efficiently. The equations are as follows:

$$x_t^{Conv1} = \sigma\left(BatchNorm\left(Conv_1^{1\times1}(x_t)\right)\right) \tag{5}$$

$$x_t^{Conv2} = Conv_2^{1\times1}\left(x_t^{Conv1}\right) \tag{6}$$

$$x_t^{Conv3} = Conv_3^{3\times3}\left(x_t^{Conv1}\right) \tag{7}$$

$$x_t^{Conv4} = Conv_4^{3\times3}\left(x_t^{Conv1}\right) \tag{8}$$

$$x_t^{Concat} = Concat\left(x_t^{Conv2}, x_t^{Conv3}, x_t^{Conv4}\right) \tag{9}$$

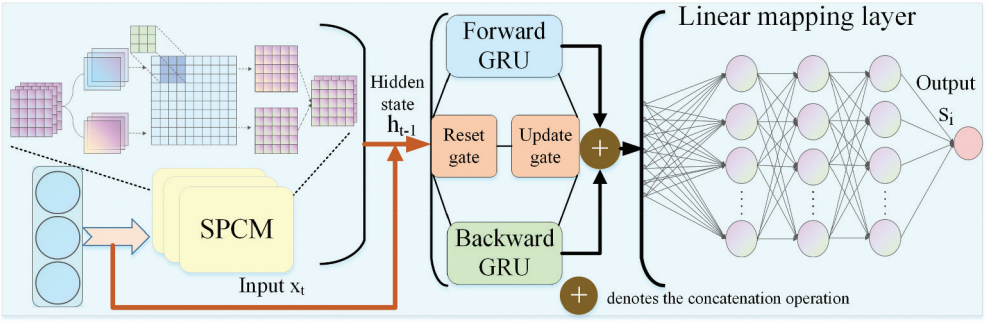


Figure 3. The overall structure of Bi-GRCU.

$$x_t^{Conv5} = \sigma\left(BatchNorm\left(Conv_5^{1\times1}\left(x_t^{Concat}\right)\right)\right) \tag{10}$$

$$h_{t-1} = x_t \oplus x_t^{Conv5} \tag{11}$$

σdenotes the ReLU function, *BatchNorm* denotes normalisation operation, *Concat* denotes the concatenation operation.

The data dimension is changed and feature extraction is performed to form h_{t-1}. Then, the input sequence x_t and h_{t-1} are simultaneously processed in the forward GRU and backward GRU, and their hidden states are connected to form a comprehensive representation. The equation is as follows:

$$h_{tt} = Concat\left(h_{ft}, h_{bt}\right) \tag{12}$$

h_{tt} denotes the final hidden state output, h_{ft}, h_{bt} are the hidden state vectors for the forward and backward GRUs output, respectively, *Concat* represents the concatenation operation.

Finally, the output is processed through a linear mapping layer for prediction. As shown in Equation (13)

$$s_i = W_s h_{tt} + b_s \tag{13}$$

$W_s \in R^{2m*d}$ is a weight matrix, $b_s \in R^d$ is a bias. Figure 3 shows the overall structure of Bi-GRCU.

Overall, Bi-GRCU addresses the limitations of traditional unidirectional recurrent networks by combining bidirectional processing with gated convolution mechanisms. This new structure enhances local context sensitivity and effectively captures short-term dependencies in sequence data, improving prediction performance.

2.2. Sparse Transformer

This article improves on the standard Transformer by proposing the Sparse Transformer, which proposes a new sparse strategy called Long-Range Locality Sparse (LRLS) Attention. LRLS-Attention is embedded in the Transformer's encoder, allowing the model to assign different weights to various positions in the input data based on their relationships with other positions. Additionally, the Sparse Transformer uses the

selective attention mechanism that significantly reduces computational and storage costs in time series prediction by focusing only on key relationships and ignoring less impactful ones, ultimately reducing data complexity.

2.2.1. Long-range locality sparse attention (LRLS-Attention)

The basic idea of LRLS-Attention is to consider two important parts: sparse window and long-range jumps. LRLS-Attention will perform sparse operations within a sparse window. Additionally, LRLS-Attention also will jump to certain distant positions at fixed distances from each position to perform sparse operations. These distant positions can be selected at regular intervals. Figure 4 shows the sparsity operation of Q and K matrices:

The sparse operation on sparse windows is as follows: given an input sequence, attention values are calculated by considering only other positions within a fixed window size, the input sequence is first weighted to form three matrices: query Q, key K and value V. The equations are as follows:

$$Q = W_q x_{ij} \tag{14}$$

$$K = W_k x_{ij} \tag{15}$$

$$V = W_V x_{ij} \tag{16}$$

W_q, W_k, W_v are weight matrices, x_{ij} is the input sequence of row i and column j.

Then, the Q and K matrices undergo similarity computation to filter irrelevant information. Subsequently, the Q and K matrices enter into the sparse operation. Before sparsity, it is necessary to set the maximum number of attention weights between each Q matrix and other K matrices and pay attention to them in order in the sparse window, that is sparsity degree b. Next, corresponding values are filled into the appropriate positions in the sparse matrix, while other positions are 0, ultimately forming the Q, K sparsity matrix. The equations of Q and K matrix are as follows:

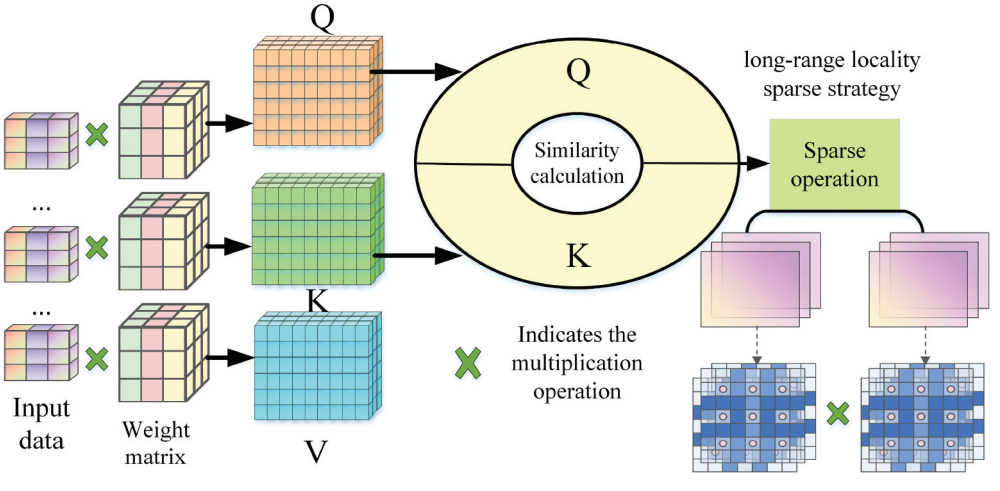


Figure 4. Sparse operation of Q and K matrices.

$$Q = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & q_{22} & \cdots & q_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ q_{m1} & q_{m2} & \cdots & q_{mn} \end{bmatrix} \tag{17}$$

$$K = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ k_{m1} & k_{m2} & \cdots & k_{mn} \end{bmatrix} \tag{18}$$

Then introduce the weight vector $w = [\, w_1, \quad w_2, \quad \ldots, \quad w_n \,]$. It assigns different weights to each column of elements. In the Q and K matrix, the sparse operation needs to find the first b elements with the largest weight, so it is necessary to design an objective function to solve the problem.

First, the sparse method defines the weighted Q, K matrix and use the nonlinear activation function to obtain the weighted nonlinear sum of each row *i*, the equations are as follows:

$$\begin{cases} f(x) = \max(0, x) \\ f(q_{ij}) = ReLU(q_{ij}) \\ S_i^q = \sum_{j=1}^{n} f(q_{ij}) w_j \end{cases} \tag{19}$$

$$\begin{cases} f(x) = \max(0, x) \\ f(k_{ij}) = ReLU(k_{ij}) \\ S_i^k = \sum_{j=1}^{n} f(k_{ij}) w_j \end{cases} \tag{20}$$

f(x) retains only positive values and suppresses negative values, S_i^q, S_i^kdenote the weighted nonlinear sum of Q, K matrices.

In order to avoid overfitting, a regularisation term R(w) is introduced and the L2 norm regularisation is selected, the equation is as follows:

$$R(w) = \lambda ||w||_2^2 = \lambda \sum\nolimits_{j=1}^{n} {w^2}_j \tag{21}$$

$\lambda = 0.01$, which is the regularization coefficient used to control the strength of regularization. λ is determined by grid search experiments.

The goal of each row *i* is to find the first b weighted nonlinear sum that maximize and to control the size of the weight vector during optimization. The final objective function can be expressed as Equations (22–23).

$$Objective_i^q = \sum\nolimits_{j=1}^{n} f(q_{ij}) w_j - \lambda ||w||_2^2 \tag{22}$$

$$Objective_i^k = \sum\nolimits_{j=1}^{n} f(k_{ij}) w_j - \lambda ||w||_2^2 \tag{23}$$

By solving the following optimisation problem: the Equations (24–25), the sparse operation gets the b elements with the largest weighted nonlinear sum in each row.

$$w_j^{q*} = Sort_b^q\left(\sum_{j=1}^{n} f\left(q_{ij}\right) w_j - \lambda \sum_{j=1}^{n} w_j^2\right) \tag{24}$$

$$w_j^{k*} = Sort_b^k\left(\sum_{j=1}^{n} f\left(k_{ij}\right) w_j - \lambda \sum_{j=1}^{n} w_j^2\right) \tag{25}$$

$$S_i^{sorted\,q} = Sort_b^q\left(f\left(q_{ij}\right) w_j^{q*}\right) \tag{26}$$

$$S_i^{sorted\,k} = Sort_b^k\left(f\left(k_{ij}\right) w_j^{k*}\right) \tag{27}$$

The Equations (24–25) can effectively control the size of the weight vector in the optimisation process and prevent the occurrence of excessive weight. $S_i^{sortedq}, S_i^{sortedk}$ denote the elements corresponding to the b largest weighted nonlinear values in each row of the Q, K matrices respectively. *Sort* refers to the sorting function.

Substitute the sparse matrices of Q and K into Equation (28) to form the local sparse attention, which is used to calculate the local mutual relationship:

$$Attention_{local}^{sparse}\left(\bar{Q}_{ij}, \bar{K}_{ij}, V_{ij}\right) = softmax\left(\frac{\bar{Q}_{ij}\bar{K}_{ij}^T}{\sqrt{d_k}}\right) V_{ij} \tag{28}$$

The $\bar{Q}_{ij}, \bar{K}_{ij}$ are Q and K matrices representing sparse processing. T represents transpose operations.

For each position i, in addition to the nearby positions within the sparse window W, LRLS-Attention will also jump to remote positions at fixed intervals k, considering remote positions $i + k$ and $i - k$. The jump interval k is a pre-set constant. The equation for calculating remote jump attention is as follows:

$$Attention_{long-range}^{sparse}\left(\bar{Q}_{ij}, \bar{K}_{ij}, V_{ij}\right) = softmax\left(\frac{\bar{Q}_{ij}\,\bar{K}_{ij}^T}{\sqrt{d_k}}\right) V_{ij} \tag{29}$$

Finally, LRLS-Attention combines local sparse attention and remote jump attention, the equation is as follows:

$$Attention_{LRLS}\begin{cases} Attention_{local}^{sparse}\left(\bar{Q}_{ij}, \bar{K}_{ij}, V_{ij}\right) & if\ j \in W_i \\ Attention_{long-range}^{sparse}\left(\bar{Q}_{ij}, \bar{K}_{ij}, V_{ij}\right) & if\ j \in \{i+k, i-k\} \\ 0 & otherwise \end{cases} \tag{30}$$

Among them, W_i is the sparse window of position i,the size is 30 ×14 which contains w positions on both sides of the position. For remote jump, position j is $i + k$ or $i - k$, and the attention score of other positions is 0. If the length of the input sequence be N, the jump interval is k, and the window length is w. The complexity analysis of each part is as follows: Each position i only considers local windows of size $2w + 1$, so the complexity of local sparse attention calculation is $O(wk)$, when each position i only needs to consider two remote positions: $i + k$ and $i - k$, the computational complexity of remote jump attention is $O(N/k)$, thus the overall complexity of LRLS-Attention is $O(wk + N/k)$. Compared with the standard fully connected self-attention (the complexity is $O(N^2)$), LRLS-Attention has

obvious computational advantages. Figure 5 shows the comparison between fully connected SAM and LRLS-Attention.

The equations of multi-head LRLS-Attention are:

$$MultiHead(\bar{Q}, \bar{K}, V) = Concat(head_1, head_2, head_3, \ldots, head_i)W^0 \tag{31}$$

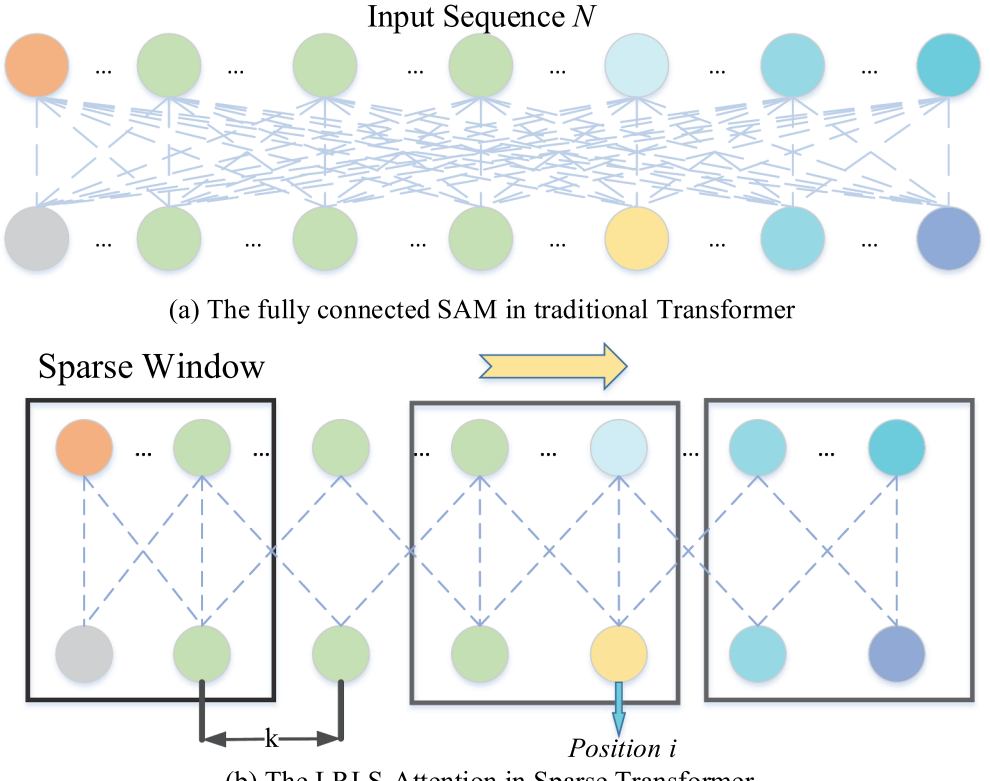
$$head_h = Attention^{sparse}\left(\bar{Q}^{(h)}, \bar{K}^{(h)}, V^{(h)}\right), h = 1, 2, 3, \ldots, h \tag{32}$$

Concat denotes the concatenation operation, and W^0 denotes the weight matrix, h denotes the number of heads. Figure 6 is a schematic process of the LRLS-Attention mechanism.

2.2.2. Position encoding

Traditional Transformers have no fixed order compared to other models, so the role of positional encoding in the model is to provide information on the various positions of the input sequence [18,19]. Thus, the Transformer can identify the relative position information of each data point.

The position encoding is represented by P_t, the time step is t, d represents the input data dimension of the Sparse Transformer. The position encoding equation is as follows:



(a) The fully connected SAM in traditional Transformer

(b) The LRLS-Attention in Sparse Transformer

Figure 5. The comparison between fully connected SAM and LRLS-Attention.

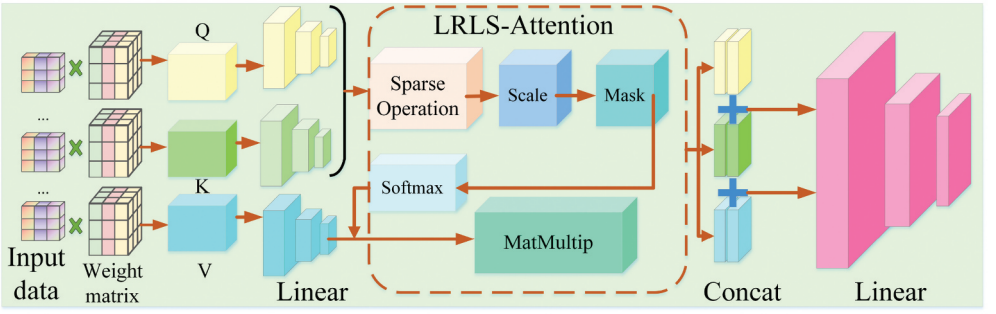


Figure 6. Schematic process of multi-head LRLS-Attention.

$$P_t^{2s} = \sin\left(t/10000^{2s/d}\right)$$
$$P_t^{2s+1} = \cos\left(t/10000^{2s/d}\right) \tag{33}$$

2.2.3. Feedforward layer

Generally speaking, the encoder structure of Transformer consists of multiple layers of stacked attention layers and fully connected layers [17]. In the Sparse Transformer, it is composed of a residual connection module, a multi-head LRLS-Attention layer and a feedforward layer. Figure 7 shows the structure of Sparse Transformer:
the equation of the feedforward layer is as follows:

$$F_{MH} = LayerNorm(MH(F) + F) \tag{34}$$

F_{MH} represents the output feature of multiple LRLS-Attention layers and the MH represents the mapping relationship of multi-head LRLS-Attention mechanism. F feature vector contains positional encoding information. $LayerNorm$ represents normalisation operation. F_{MH} passes through the feedforward layer, it can obtain F_{FF}, which is the output of the encoding layer. F_{FF} is shown in Equation (35):

$$F_{FF} = LayerNorm(f_{FF}(F_{MH}) + F_{MH}) \tag{35}$$

f_{FF} represents the output of the feedforward layer is shown in Equation (36):

$$f_{FF}(F_{MH}) = f_{ReLU}(W_{FF}F_{MH} + b_{FF})$$
$$f_{ReLU}(x) = max\{0, x\} \tag{36}$$

W_{FF} and f_{ReLU} are the weight matrix and b_{FF} is bias matrix, $ReLU$ is activation function. Processing the data through the feedforward layer helps alleviate the problem of gradient vanishing, enabling the model to train better and effectively improve computational efficiency.

2.3. RUL prediction process based on the proposed model

After feature extraction and calculation using Bi-GRCU and Sparse Transformer, the raw data x_t enters the linear regression layer and ultimately produces the predicted RUL. The equation involved in the linear regression layer is as follows:

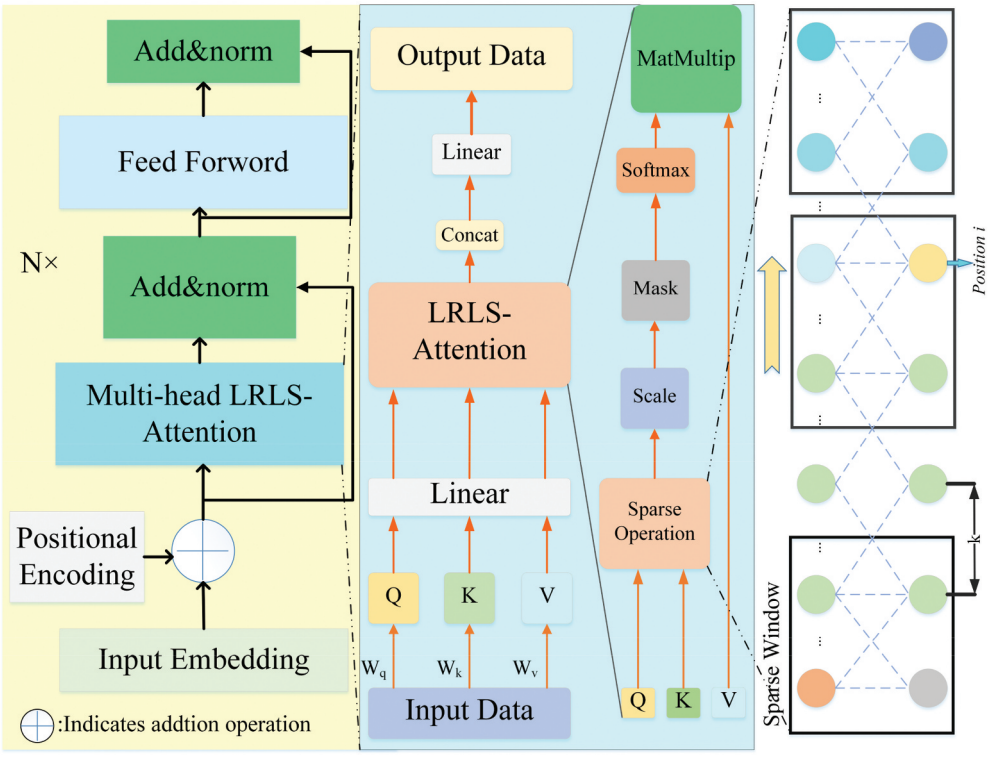


Figure 7. Sparse transformer structure.

$$y_t^r = sigmoid(W_0 g_t + b_0) \tag{37}$$

$W_0 \in R^d$, b_0 is a scalar, g_t is the output of Sparse Transformer and y_t^r is the predicted value of RUL.

When performing RUL prediction tasks, the pre-trained model is first trained to obtain the optimal model parameters and optimiser state, then these parameters and state are saved. This allows the trained parameters to be directly used for prediction, so that the prediction results can be quickly obtained and greatly improve prediction efficiency. Figure 8 shows the experimental process of the proposed model.

3. RUL prediction and comparison of aircraft turbofan engines

3.1. CMAPSS dataset

The CMAPSS dataset was developed by NASA to promote research and evaluation of aircraft engine performance monitoring algorithms. Figure 9 shows the simulated structure of the aeroengine.

The CMAPSS dataset has four sub datasets: FD001-FD004, they mainly simulate various types of aircraft engines and each sub dataset contains a lot of sensor data. The engines in these datasets experienced different types and degrees of

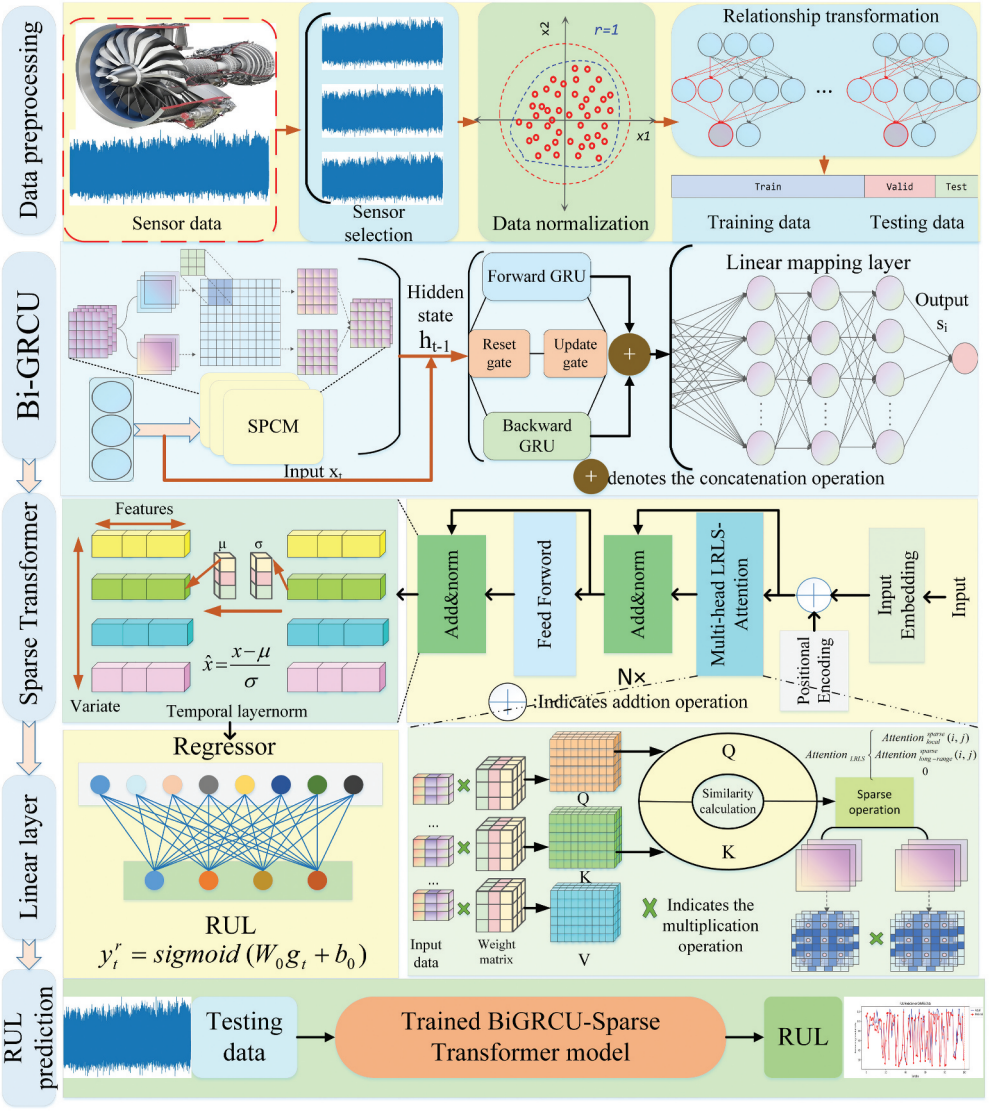


Figure 8. The experimental process of the proposed model.

wear and failure during the simulation process, which provided abundant samples for model training and validation [20]. This paper conducts experiments using four subsets of CMAPSS. Table 1 describes the specific details of the CMAPSS dataset:

3.1.1. Sensor selection

Each subset in the CMAPSS dataset contains three operation settings, time step, engine number and 21 sensor measurement data, which can provide degradation information for turbofan engines. However, it can be observed that as the lifespan increases, some of the sensor samples do not show significant changes. These data

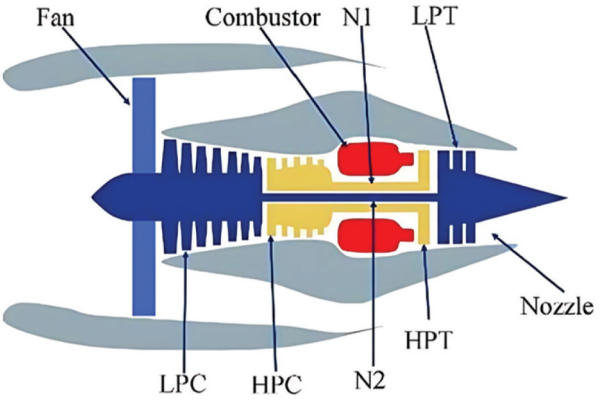


Figure 9. The simulated structure of the aeroengine.

Table 1. The CMAPSS dataset Description.

	CMAPSS dataset			
Dataset	FD001	FD002	FD003	FD004
Fault mode	1	1	2	2
Operating conditions	1	6	1	6
Training engine	100	260	100	249
Test engine	100	259	100	248

cannot provide useful information about rule learning and will increase the complexity of neural networks.

Reference [21] indicates that out of a total of 21 sensors, data from 14 sensors can be used for RUL prediction, with metrics of 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20 and 21, the reason is that the sensors are highly correlated with each other.

The correlation quantification equation of the sensor is as followed in Equation (38). Therefore, in this article, the 14 sensors data mentioned above will be used for RUL prediction.

$$\rho_{X,Y} = \frac{COV(X,Y)}{\sigma_X \sigma_Y} = \frac{E\big((X-\mu_X)(Y-\mu_Y)\big)}{\sigma_X \sigma_Y} \tag{38}$$

σ_X, σ_Y represents the standard deviation of X and Y, respectively, $\rho_{X,Y}$ denotes the correlation between Sensor X and Sensor Y. Figures 10 and 11 respectively display the engine correlation matrices for the CMAPSS training and testing sets.

3.1.2. Data normalization

After selecting the sensors, the data must be normalized prior to being fed into the neural model to perform tasks. In this sense, this article applies max-min normalization to preprocess the data, processing its values into the range of [0,1], thereby reducing data complexity and improving computational efficiency for the next prediction task. As shown in Equation (39):

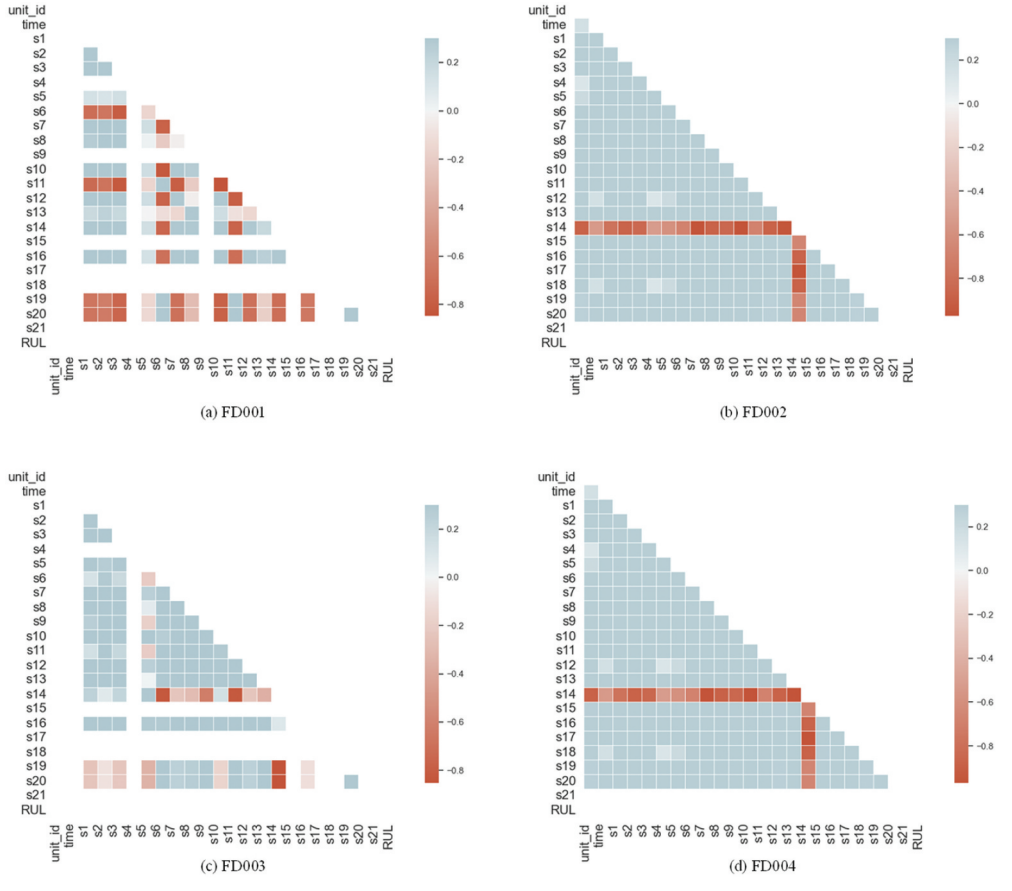


Figure 10. The engine correlation matrix in CMAPSS training set.

$$x_{norm}^{i,j} = \frac{x^{i,j} - x_{min}^{j}}{x_{max}^{j} - x_{min}^{j}} \tag{39}$$

$x_{norm}^{i,j}$ represents the normalized value, $x^{i,j}$ represents the i raw data j sensor after sensor selection, and x_{max}^{j}, x_{min}^{j} represent the highest and lowest values in the sensor's raw data.

3.2. Predictive performance evaluation indicators

Two indicators are used for predictive performance evaluation in this article, which are root mean square error (RMSE) and S-Score. The RMSE and S-Score are shown in Equations (40–41):

$$RMSE = \sqrt{\frac{\sum x \in n_{test}(y_t - y_t^r)^2}{n_{test}}} \tag{40}$$

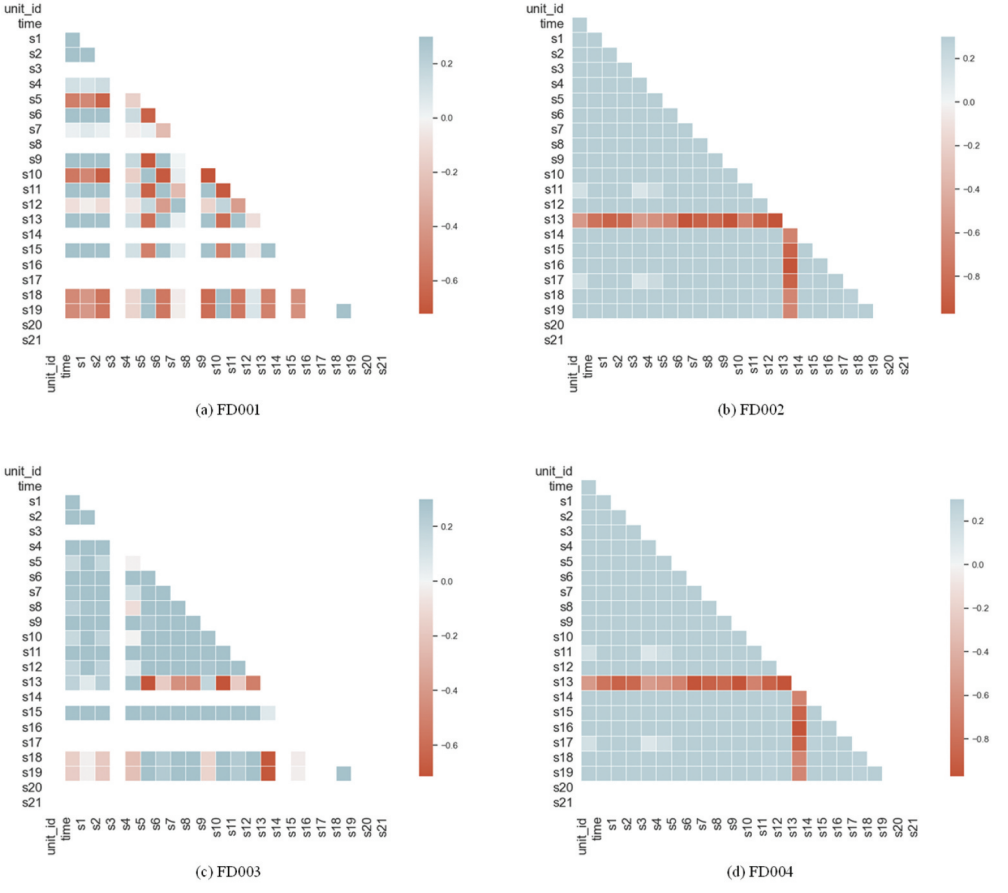


Figure 11. The engine correlation matrix in CMAPSS testing set.

$$S = \begin{cases} \sum_{x=1}^{n_{test}} exp^{-\left(\frac{y_t^r - y_t}{13}\right)} - 1 \\ \sum_{x=1}^{n_{test}} exp^{\left(\frac{y_t^r - y_t}{10}\right)} - 1 \end{cases} \tag{41}$$

The values of the RMSE and S-Score are obtained by substituting the full number of test sets n_{test}, the predicted RUL y_t^r and the real RUL y_t, into Equations (40–41) are used to measure the predictive performance of the proposed model. The smaller the values of the RMSE and S-Score, the stronger the predictive ability.

3.3. Experimental details and results

3.3.1. Experimental details

In this paper, the FD001 data subset is used to control a single variable for experimental verification, so as to determine the main hyperparameters. Figure 12 shows the influence of different sparsity degrees, time windows, encoder layers and the head numbers of LRLS-attention mechanism on the prediction results.

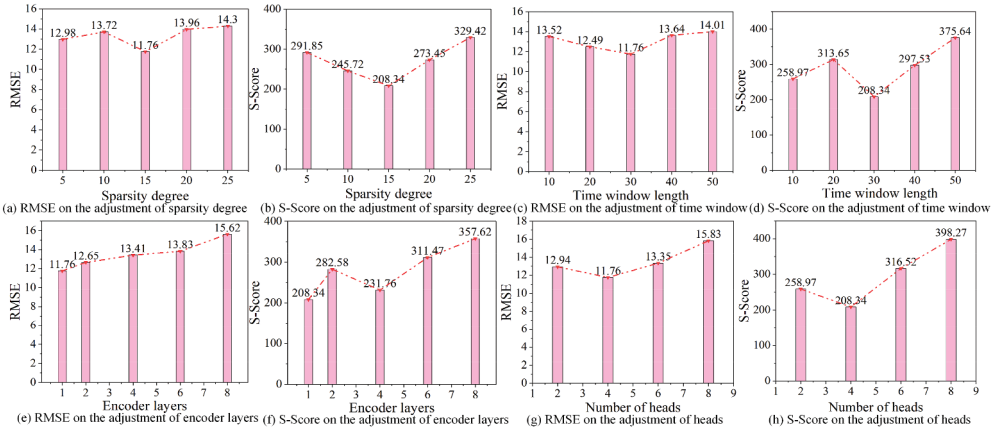


Figure 12. The hyperparameter comparison experiment based on FD001.

Based on experience process the dataset into $k \times m$ input data, set the count of encoder layers to $N = 1$, the head count of the sparse attention is 4, the learning rate is 0.001 and input the data dimension to Sparse Transformer is 64. Due to the fact that the original data volume of the FD001 and FD003 subsets in the aeroengine dataset is roughly the same, the value of the Dropout is 0.1 during the experiment. The original data volume of FD002 and FD004 is roughly the same, thus the value of the Dropout is 0.5 during the experiment. In the experiment, the number of rounds of the model is 20. Table 2 shows the model parameter configuration. Table 3 shows the data structure of the model:

3.3.2. Experimental results

After setting the hyperparameters according to Table 2, this paper conducted experiments on four subsets of the CMAPSS dataset. The experimental results obtained proved that the proposed model has satisfactory predictive ability. Figure 13 shows the visualization results of the RUL of four data subsets after performing prediction tasks.

According to Table 4, the values of the RMSE and S-Score calculated by other models are compared with the model proposed in this paper. It is evident that the model has

Table 2. Model parameter configuration.

Hyperparameters	values
Input dimension *m*	14
Encoder layers *N*	1
Number of heads *h*	4
Learning rate	0.001
Input to encoder dimension *d*	64
Dropout value	0.1,0.5
jump interval *k*	30
optimizer	Adam
loss function	Mean square error
Sparsity degree *b*	15
Training rounds	20

Table 3. Data structure of the model.

Module	Layer	Size
Input		(1,30,14)
Feature extractor	Bi-GRCU	(1,30,64)
	Sparse Transformer	(1,30,64)
Regressor	Linear regression	(1,10)
Output		(1,1)

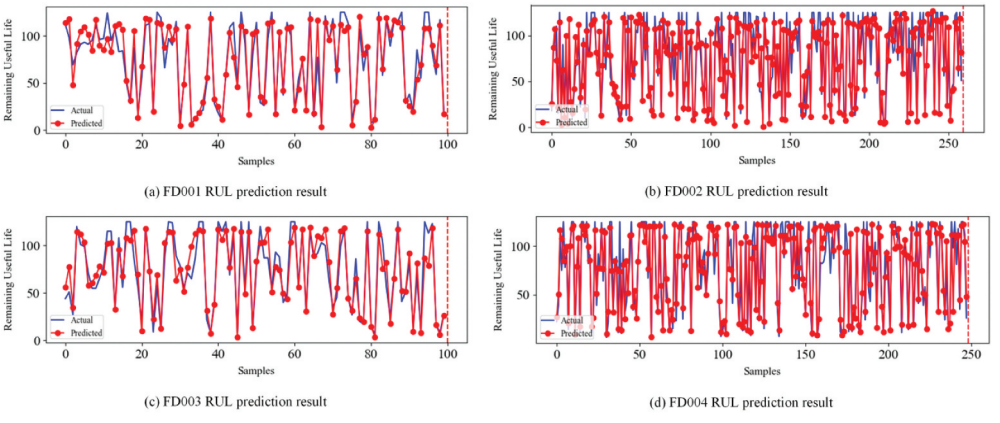


Figure 13. FD004 RUL prediction results.

Table 4. Evaluation of RUL prediction performance.

	RMSE				S-Score			
Approach	FD001	FD002	FD003	FD004	FD001	FD002	FD003	FD004
ELM [22]	17.27	37.28	18.90	38.43	523.00	–	573.78	–
CNN+RNN [23]	16.89	30.97	17.82	29.73	820.67	15917.00	950.94	7212.20
CNN+LSTM [6]	16.13	20.46	17.12	23.26	303.00	3440.00	1420.94	4630.00
CEED+DLSTM [24]	14.72	29.00	17.72	33.43	262.00	6953.00	452.00	15069.00
Bi-LSTM-ED [25]	14.74	22.07	17.48	23.49	273.00	3099.00	574.00	3202.00
IESGP [26]	14.72	24.81	14.99	28.61	331.90	4245.40	355.20	6280.80
CNN-LSTM-DA [27]	14.40	27.23	14.32	26.69	290.00	9869.00	316.00	6594.00
ABGRU [28]	–	17.97	–	21.55	–	2072	–	3626
Informer [29]	13.65	14.58	12.99	14.81	285.00	1040.00	261.00	**1071.00**
Bi-GRU-TSAM [30]	12.56	18.94	12.45	20.47	213.35	2264.13	232.86	3610.34
TATFA-Transformer [31]	12.21	15.07	**11.23**	18.81	261.50	1359.70	**210.21**	2506.35
IMDSSN [19]	12.14	17.40	12.35	19.78	**206.11**	1775.15	229.54	2852.81
PSTFormer [1]	12.08	13.00	12.11	14.74	224	877	308	1182
Proposed Model	**11.76**	**12.73**	11.99	**14.73**	208.34	**717.58**	221.48	1072.61
PI (%)	**+2.62**	**+2.08**	–	**+0.07**	–	**+18.18**	–	–

*PI denotes the percentage increase in the predictive performance of the proposed model compare to other best values.

excellent predictive capabilities and is superior to other existing methods. The visualization results of the RUL prediction further confirm this experimental conclusion.

4. PHM2010 RUL prediction and comparison

The effectiveness of the proposed model in RUL prediction will be further validated through experiments on the PHM2010 dataset.

4.1. Dataset details and preprocessing process

PHM2010 dataset consists of six files, including C1 to C6. C1, C4 and C6 are used as training datasets and C2, C3 and C5 are used as test datasets [32]. Each file has seven columns of data, denoting forces (N), vibrations (g) and acoustic emission (V) in the X, Y and Z directions. The experiment utilised files C1, C4 and C6 for model validation, as these were the only datasets containing complete tool wear measurements and represented operational conditions covering all failure modes present in the test set [33,34].

This article first performs feature extraction on the dataset, including time domain feature extraction, time-frequency domain feature extraction and frequency-domain feature extraction. The extracted features are then subjected to feature calculations, including absolute mean (AM), max value (Max), root mean square value (RMS), root mean square amplitude (SRA), skewness value (SK), kurtosis value (KU), waveform factor (SF), pulse factor (PF), skewness factor (SF1), peak factor (CF), clearance factor (CF1), kurtosis factor (KF), etc. These features are used to describe different aspects of the data, such as amplitude, frequency, skewness, etc. of vibration signals. Frequency domain feature extraction applies Fourier transform to the extracted data to calculate frequency domain features, including frequency FC, mean square frequency MSF, root mean square frequency RMSF, frequency variance VF, etc. These features are used to depict the distribution of signals in the frequency domain. The time-frequency domain feature extraction part uses wavelet transform to perform time-frequency domain analysis on the extracted data, obtaining information on different frequency components. These features (F1-F8) are used to describe the changes of the signal in the time-frequency domain. Figure 14 shows the extracted feature maps of C1, C4 and C6.

The experimental design employs cross-validation to partition the PHM2010 dataset: Task A uses C1 and C6 as training sets with C4 as the test set, while Task B swaps the roles of C4 and C6 for training and C1 for testing. This cross-validation approach not only evaluates the model generalization capability across different operational conditions but also ensures the reliability of experimental results. Table 5 provides a detailed list of them.

Afterwards, the signal data is normalised and input into the model, using Equation (40) RMSE and Equation (42) MAE as the performance evaluation metric.

$$MAE = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \left| y_t^r - y_t \right| \tag{42}$$

n_{test} is the full number of test sets, y_t^r is the predicted RUL, y_t is the real RUL.

4.2. Experimental process and results

According to Table 6 set the hyperparameters and process the dataset into input data x with dimensions (315, 6, 24).

Based on experience, the number of encoder layers N is 1, the head number of sparse attention mechanism is 4, input to the sparse transformer data dimension d is 64, the sparse degree is 6.

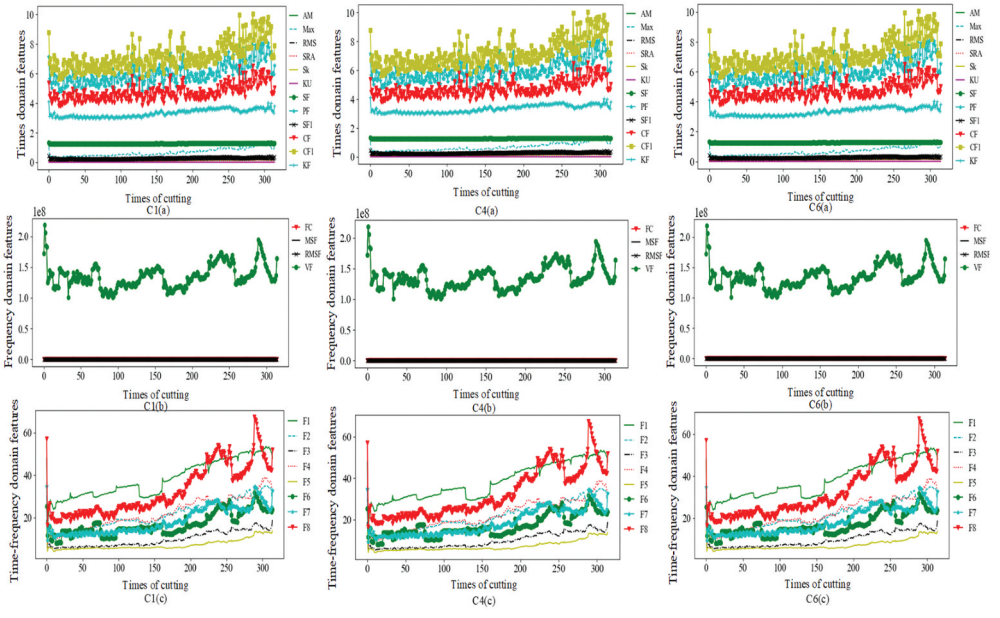


Figure 14. C1, C4 and C6 feature diagram.

Table 5. PHM2010 prediction tasks.

Task	Train	Test
A	C1, C6	C4
B	C4, C6	C1

Table 6. PHM2010 hyperparameter settings.

Hyperparameters	Hyperparameter values
Encoder layers *N*	1
Number of heads *h*	4
Data dimension *d*	64
Sparsity degree *b*	6

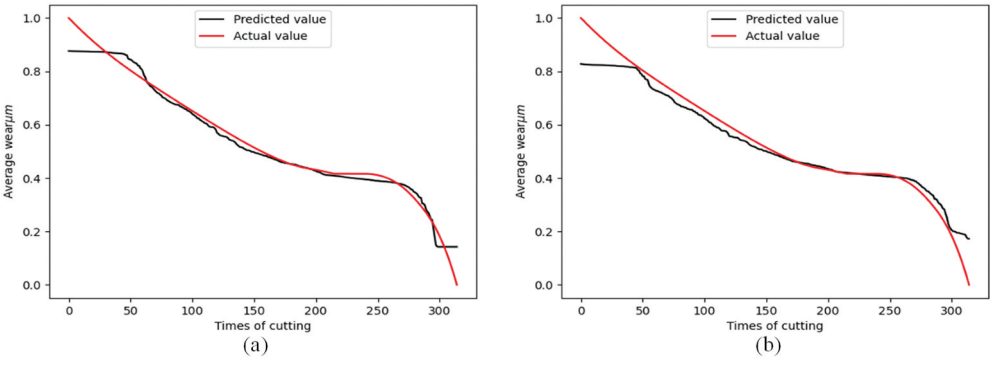


Figure 15. RUL prediction results for group A (a) and group B (b).

Table 7. RMSE and MAE values comparison.

	A		B	
Method	RMSE	MAE	RMSE	MAE
LR [33]	15.4	13.1	19.1	10.0
SVR [33]	12.7	10.6	13.3	8.6
ResNet [33]	11.7	8.8	13.6	8.9
ResNetSBiLSTM [33]	10.9	7.4	14.5	7.4
Proposed model	7.2	6.3	7.4	6.7
PI (%)	33.94	14.86	44.36	9.46

Figure 15 shows the visualization results of RUL predictions for Group A and Group B respectively. Afterwards, the RMSE and MAE values obtained will be compared with the values obtained by other deep learning-based methods. The results are shown in Table 7:

Through experimental verification, it is known that the Sparse Transformer with Bi-GRCU has excellent performance in predicting the RUL.

5. Conclusion

This paper proposes a novel sequence ensemble model, which is primarily made up of two components: Bi-GRCU and Sparse Transformer. Bi-GRCU combines characteristics of RNN and CNN, which can effectively distill important features from raw input data, enhance the perception and connection of local context in data. Compared with traditional unidirectional recurrent networks, Bi-GRCU can simultaneously use future and past information to predict the current output. In order to better capture long-term dependencies, Sparse Transformer is adopted, which incorporates the LRLS-Attention mechanism into the encoder structure of the Transformer. This mechanism assigns different weights to information positions based on their relationships with other positions, allowing the model to better extract long-term feature dependencies from the output of Bi-GRCU. In addition, Sparse Transformer significantly reduces computational and storage costs in time series prediction by focusing only on key relationships and ignoring less impactful ones. The proposed method was validated using the CMAPSS aeroengine dataset and the PHM2010 dataset, where it demonstrated superior predictive performance compared to other existing models.

However, despite its promising results, this model has certain limitations. First, the Bi-GRCU and Sparse Transformer combination may not perform optimally in highly noisy environments, as the model could be sensitive to irrelevant input features. Additionally, the complexity of the Sparse Transformer architecture can result in high memory and computational requirements, particularly when dealing with very large datasets. Future research may focus on improving the robustness of the model in noisy conditions, as well as exploring more efficient versions of Sparse Transformer to reduce resource consumption. Further work may also investigate applying this approach to other domains to explore its generalisation capabilities.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the National Natural Science Foundation of China (Nos. 52305085 and 52105111), China Postdoctoral Science Foundation (2023M740021).

Data availability statement

The data that support the findings of this study are available in the GitHub repository, and can be accessed at [GitHub – NicolasDeHorta/CMAPSSData: CMAPSS dataset from NASA turbofans] and https://phmsociety.org/phm_competition/2010-phm-society-conference-data-challenge/.

References

[1] Fu S, Jia YM, Lin L, et al. Pstformer: a novel parallel spatial-temporal transformer for remaining useful life prediction of aeroengine. Expert Syst With Appl. 2025;265:125995. doi: 10.1016/j.eswa.2024.125995
[2] Deng L, Li W, Yuan X. An intelligent hybrid deep learning model for rolling bearing remaining useful life prediction. Nondestr Test Evaluation. 2025;40(6):2670–2697. doi: 10.1080/10589759.2024.2385074
[3] Yan X, Jin X, Jiang D, et al. High-resolution ultrasonic image reconstruction of shallow rail defects using a relativistic-attention-weighted cyclegan. Nondestr Test Evaluation. 2025:1–29. doi: 10.1080/10589759.2025.2541051
[4] Yin S, Rodriguez-Andina JJ, Jiang Y. Real-time monitoring and control of industrial cyberphysical systems: with integrated plant-wide monitoring and control framework. IEEE Ind Electron Mag. 2019;13(4):38–47. doi: 10.1109/MIE.2019.2938025
[5] Wei L, Peng X, Cao Y. Enhanced fault diagnosis of rolling bearings using an improved inception-LSTM network. Nondestr Test Evaluation. 2024;40(7):3274–3293. doi: 10.1080/10589759.2024.2402549
[6] Kong Z, Cui Y, Xia Z, et al. Convolution and long short-term memory hybrid deep neural networks for remaining useful life prognostics. Appl Sci. 2019;9(19):4156. doi: 10.3390/app9194156
[7] Qi JY, Chen ZY, Song YC, et al. Remaining useful life prediction combining advanced anomaly detection and graph isomorphic network. IEEE Sensors J. 2024;24(22):38365–38376. doi: 10.1109/JSEN.2024.3470231
[8] Li X, Ding Q, Sun J-Q. Remaining useful life estimation in prognostics using deep convolution neural networks. Reliab Eng Syst Saf. 2018;172:1–11. doi: 10.1016/j.ress.2017.11.021
[9] Huang C-G, Huang H-Z, Li Y-F. A bidirectional LSTM prognostics method under multiple operational conditions. IEEE Trans Ind Electron. 2019;66(11):8792–8802. doi: 10.1109/TIE.2019.2891463
[10] Zhang J, Tian J, Li M, et al. A parallel hybrid neural network with integration of spatial and temporal features for remaining useful life prediction in prognostics. IEEE Trans Instrum Meas. 2022;72:1–12. doi: 10.1109/TIM.2022.3227956
[11] Liu J, Lei F, Pan C, et al. Prediction of remaining useful life of multistage aero-engine based on clustering and LSTM fusion. Reliab Eng Syst Saf. 2021;214:107807. doi: 10.1016/j.ress.2021.107807
[12] Shi Z, Chehade A. A dual-LSTM framework combining change point detection and remaining useful life prediction. Reliab Eng Syst Saf. 2021;205:107257. doi: 10.1016/j.ress.2020.107257

[13] Zhang Z, Song W, Li Q. Dual-aspect self-attention based on transformer for remaining useful life prediction. IEEE Trans Instrum Meas. 2022;71:1–11. doi: 10.1109/TIM.2022.3160561
[14] Li X, Li J, Zuo L, et al. Command filter-based adaptive fuzzy finite-time output feedback control of nonlinear electrohydraulic servo system. IEEE Trans Instrum Meas. 2022;71:1–10.
[15] Liu L, Song X, Zhou Z. Aircraft engine remaining useful life estimation via a double attention-based data-driven architecture. Reliab Eng Syst Saf. 2022;221:108330. doi: 10.1016/j.ress.2022.108330
[16] Zhou JH, Qin Y, Luo J, et al. Remaining useful life prediction by distribution contact ratio health indicator and consolidated memory GRU. IEEE Trans On Ind Inf. 2023;19(7):8472–8483. doi: 10.1109/TII.2022.3218665
[17] Chen DL, Qin Y, Wang Y, et al. Health indicator construction by quadratic function-based deep convolutional autoencoder and its application into bearing RUL prediction. ISA Trans. 2021;114:44–56. doi: 10.1016/j.isatra.2020.12.052
[18] Qi JY, Chen ZY, Kong Y, et al. Attention-guided graph isomorphism learning: a multi-task framework for fault diagnosis and remaining useful life prediction. Reliab Eng Syst Saf. 2025;263:111209. doi: 10.1016/j.ress.2025.111209
[19] Zhang JS, Li X, Tian JL, et al. An integrated multi-head dual sparse self-attention network for remaining. Reliab Eng And System Saf. 2023;233:109096. doi: 10.1016/j.ress.2023.109096
[20] Yu Mo QW, Li X, Huang B. Remaining useful life estimation via transformer encoder enhanced by a gated convolutional unit. J Intell Manuf. 2021;32(7):1997–2006. doi: 10.1007/s10845-021-01750-x
[21] Saxena A, Goebel K, Simon D, et al. Damage propagation modeling for aircraft engine run-to-failure simulation. In: 2008 international conference on prognostics and health management; Denver, CO, USA: IEEE; 2008. p. 1–9.
[22] Zhang C, Lim P, Qin AK, et al. Multi-objective deep belief networks ensemble for remaining useful life estimation in prognostics. IEEE Trans Neural Netw Learn Syst. 2016;28(10):2306–2318. doi: 10.1109/TNNLS.2016.2582798
[23] Zhang X, Dong Y, Wen L, et al. Remaining useful life estimation based on a new convolutional and recurrent neural network. 2019 IEEE 15th Int Conf On Automation Sci Eng (case). 2019: 317–322.
[24] Guo J, Li DP, Du BG. A stacked ensemble method based on TCN and convolutional bi-directional GRU with multiple time windows for remaining useful life estimation. Appl Soft Comput. 2024;150:111071. doi: 10.1016/j.asoc.2023.111071
[25] Yu W, Kim IY, Mechefske C. Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. Mech Syst Signal process. 2019;129:764–780. doi: 10.1016/j.ymssp.2019.05.005
[26] Liu C, Zhang L, Liao Y, et al. Multiple sensors based prognostics with prediction interval optimization via echo state Gaussian process. IEEE Access. 2019;7:112397–112409. doi: 10.1109/ACCESS.2019.2925634
[27] Wu Z, Yu S, Zhu X, et al. A weighted deep domain adaptation method for industrial fault prognostics according to prior distribution of complex working conditions. IEEE Access. 2019;7:139802–139814. doi: 10.1109/ACCESS.2019.2943076
[28] Wang J, Lu Z, Zhou J, et al. A novel remaining useful life prediction method under multiple operating conditions based on attention mechanism and deep learning. Adv Eng Inf. 2025;64:103083. doi: 10.1016/j.aei.2024.103083
[29] Wang J, Wen G, Yang S, et al. Remaining useful life estimation in prognostics using deep bidirectional LSTM neural network. In: 2018 Prognostics and system health management conference ;Chongqing, PEOPLES R CHINA: IEEE; 2018. p. 1037–1042.
[30] Zhang J, Jiang Y, Wu S, et al. Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism. Reliab Eng Syst Saf. 2022;221:108297. doi: 10.1016/j.ress.2021.108297

[31] Zhang Y, Su C, Wu J, Liu H, Xie M. Trend-augmented and temporal-featured transformer network with multi-sensor signals for remaining useful life prediction. Reliab Eng Syst Saf. 2024;241:109662. doi: 10.1016/j.ress.2023.109662
[32] Society P. Phm society conference data challenge [online]. 2010. Available from: https://phmsociety.org/phm_competition/2010-phm-society-conference-data-challenge/
[33] Liu X, Liu S, Li X, et al. Intelligent tool wear monitoring based on parallel residual and stacked bidirectional long short-term memory network. J Manuf Syst. 2021 Jul;60:608–619. doi: 10.1016/j.jmsy.2021.06.006
[34] Wei LP, Peng XY, Cao YP. Remaining useful life prediction of rolling bearings using a residual attention network with multi-scale feature extraction and temporal dependency enhancement. Nondestr Test Evaluat. 2025:1–23. doi: 10.1080/10589759.2025.2451772