Enhancing Remaining Useful Life Prediction with Ensemble Multi-Term Fourier Graph Neural Networks

Anonymous authors
Paper under double-blind review

Abstract

Remaining useful life (RUL) prediction is crucial in predictive maintenance. Recently, deep learning forecasting methods, especially Spatio-Temporal Graph Neural Networks (ST-GNNs), have achieved remarkable performance in RUL prediction. The existing ST-GNNs require searching for the graph structure before utilizing GNNs to learn spatial graph representation, and they necessitate a temporal model such as LSTM to leverage the temporal dependencies in a fixed lookback window. However, such an approach has several limitations. Firstly, it demands substantial computational resources to learn graph structures for the time series data. Secondly, independently learning spatial and temporal information disregards their inherent correlation, and thirdly, capturing information within a fixed lookback window ignores long-term dependencies across the entire time series. To mitigate the issues above, instead of treating the data within the lookback window as a sequence of graphs in ST-GNN methods, we regard it as a complete graph and employ a Fourier Graph Neural Network (FGN) to learn the spatiotemporal information within this graph in the frequency space. Additionally, we create training and test graphs with varying sizes of lookback windows, enabling the model to learn both short-term and long-term dependencies and provide multiple predictions for ensemble averaging. We also consider scenarios where sensor signals exhibit multiple operation conditions and design a sequence decomposition plugin to denoise input signals, aiming to enhance the performance of FGN. We evaluate the proposed model on the C-MAPSS benchmark dataset, demonstrating its superior performance on the RUL prediction task compared to state-of-the-art approaches.

1 Introduction

The widespread adoption of Cyber-Physical Systems (CPS) and the Internet of Things (IoT) has enabled organizations to leverage predictive maintenance techniques more effectively. This has resulted in extended equipment operational lifecycles, prevented unscheduled downtime, and decreased energy consumption. Predictive maintenance is a proactive approach that estimates the equipment's Remaining Useful Life (RUL), i.e., forecasts the time point at which equipment may fail or become ineffective in future usage. It then develops appropriate maintenance plans and procedures to ensure the equipment's reliability and continuous operation. Among these, forecasting RUL is considered the most significant and valuable task Zhou et al. (2021). RUL prediction typically depends on historical operational data and condition monitoring information, including sensor data, operation records, maintenance histories, and so on. Common prediction approaches include those based on physical models, data-driven methods, and hybrid methods Ferreira & Gonçalves (2022). These models examine equipment's operational status and health to predict probable failures. With the advancements in sensing technology and data analytics, data-driven strategies, especially deep learning-based RUL prediction, are emerging as a significant research and application area in engineering.

When utilizing deep learning techniques for RUL prediction, we tackle the problem as a multivariate time series regression task. Generally, we apply the sliding time window approach to generate samples of time series data. The basic idea is to segment raw time series data by sliding a fixed-length time window, taking the data within each time window as a sample, and then using these samples for machine learning model training and prediction. Existing deep learning models for predicting RUL mostly use Convolutional Neural Networks (CNNs) Yang et al. (2019); Ren et al. (2020); Xu et al. (2022), Long Short-Term Memory (LSTMs) Shi & Chehade (2021); Wu et al. (2021), and Transformers Li et al. (2022); Zhang et al. (2022b); Jiang et al. (2023). These methods have effectively captured the temporal dependencies in time series data. However, they are limited as they do not consider the potential interactions between sensor signals, hindering prediction models' effectiveness. Researchers have started using Spatio-Temporal Graph Neural Networks (ST-GNNs) Jin et al. (2023) to address this problem. This approach constructs a graph for the sensor signal data at each time step, leveraging GNNs to capture spatial information within the graph, followed by sequence models such as LSTMs to abstract the temporal information of a sequence of graph embedding.

Although the experimental results in Kong et al. (2022); Wang et al. (2023a; 2024) show ST-GNNs can outperform traditional sequence models in RUL prediction tasks, the current ST-GNNs models still have the following four main drawbacks: (1)*Need to learn the graph structure.* Unlike the natural graph structure inherent in problems like traffic flow prediction in road networks, there is no explicit graph structure among sensor signals in RUL prediction. Therefore, it often necessitates the use of graph structure learning methods, demanding substantial computational resources. (2)*Modeling spatial and temporal information separately.* The conventional ST-GNNs separately employ GNN to capture spatial information and LSTM to capture temporal information. This technique fails to consider the possible spatiotemporal inter-dependencies present in sensor signals. (3)*Fixed and short-term dependency modelling.* Traditional models employ a single fixed lookback window to generate samples, often resulting in a window size too small for longer time series data. Consequently, the model struggles to capture long-term dependencies across the entire time series. (4)*Ignore operation condition information.* Equipment may operate under various operation conditions, and analyzing historical operational records can facilitate the learning of potential degradation trends. Existing models solely focus on modeling sensor signals, disregarding operation condition information.

To tackle the above problems, we propose an RUL prediction model named Ensemble Multi-Term Fourier Graph Neural Networks (MT-FGNE). The characteristics and advantages of this model are as follows:

- We adopt a novel approach to time series processing. We no longer view a sample as a sequence of graphs like ST-GNNs; instead, we consider samples as complete graphs. After converting it to the frequency domain using the Discrete Fourier Transform (DFT), we utilize Fourier Graph Neural Networks (FGN) to capture the degradation trends. This approach avoids separately modeling spatial and temporal information, enabling the learning of potential spatiotemporal interdependencies within sensor signal data.
- We propose a multi-term learning framework to address the issue of traditional models' inadequate learning of long-term dependencies. In this framework, we generate training and test graphs with variable lookback window sizes, allowing the model to learn both short-term and long-term dependencies while providing multiple predictions for ensemble averaging.
- We consider the specificity of degradation under multiple operation conditions by incorporating historical operational recording data into the modeling process. After decomposing and interpolating the original signals, we input them into the model and average the prediction results.
- We evaluate our MT-FGNE model on widely studied benchmarks and achieve competitive performance against state-of-the-art traditional and ST-GNN methods.

2 Related Work

2.1 Deep learning models for RUL prediction

Due to their ability to handle complex nonlinear relationships and perform end-to-end learning, deep learning models have been applied to RUL prediction tasks for quite some time. CNNs can extract local patterns from time series data, while LSTMs effectively capture long-term dependencies by introducing gate units. The early models consisted mainly of CNNs Yang et al. (2019); Ren et al. (2020); Xu et al. (2022) and LSTMs Shi & Chehade (2021); Wu et al. (2021). Hybrid models combining both have also been widely utilized, leveraging their capability to extract both spatial and temporal features Zraibi et al. (2021); Ren et al. (2020). Through self-attention mechanisms, Transformers can attend to all positions in the input sequence at each time step, thus better capturing long-term dependencies and global patterns in time series data. Researchers have also applied them to RUL prediction and made various improvements to the attention mechanism Li et al. (2022); Zhang et al. (2022b); Jiang et al. (2023).

Later, researchers recognized that GNNs are superior tools for modeling spatial information. They began constructing a graph from the signal data at each time step and employed graph neural networks for spatial feature learning. The main GNN architectures include Graph Convolutional Network (GCN) Wang et al. (2021; 2023a), Graph Attention Network (GAT) Zhang et al. (2022b); Kong et al. (2022), and custom Message Passing Neural Networks (MPNNs) Wang et al. (2023b). In terms of graph construction, one study generates graphs based on domain knowledge Kong et al. (2022), while another calculates adjacency matrices based on Pearson Correlation Coefficients among sensors Wang et al. (2021). Recent research prefers to apply graph structure learning approaches. In Chen & Zeng (2023), the authors construct the graph structure by computing the cosine similarity between the embedding vectors outputted by GAT. In Wang et al. (2023a), researchers design a dynamic graph learning module to capture the dynamic relationships between sensor data. CDSG Wang et al. (2023a) examines how time scales impact predictions by dividing the data into segments inside a lookback window, generating several time scales to provide more detailed structural insights.

In summary, existing ST-GNN methods commonly view samples as sequences of graphs, often requiring graph construction and learning on the original data in the time domain. The FGN method employed in this paper differs significantly in treating samples as complete graphs, eliminating the need to learn graph structures, and transforming the samples into the frequency domain through DFT.

2.2 Multi-term learning

The closest idea to our proposed multi-term learning is multi-scale learning. Multi-scale learning is commonly employed in a variety of domains and tasks, including computer vision and time series data analysis. By employing multi-scale learning, CNNs can extract features from receptive fields of different sizes simultaneously, enabling a more comprehensive capture of information in images, including both local details and global structures Cai et al. (2016). CrossViT Chen et al. (2021a) divides input images into multiple patches of different sizes and employs a multi-scale feature fusion mechanism to integrate feature representations from different scales. As time series data typically contains patterns and trends at multiple time scales, adopting multi-scale learning in time series analysis helps models gain a more comprehensive understanding and capture structural information within the data Cui et al. (2016). A classical approach is to apply multi-scale convolution, which generates feature maps at different scales to capture information along the time axis Chen & Shi (2021); Chen et al. (2021b). In Chen et al. (2023), researchers apply a multi-scale pyramid network to preserve the various temporal dependencies. The model's input remains at equal scales in the aforementioned multi-scale methods. In contrast, our multi-term learning approach generates inputs of varying scales by employing different lookback window sizes. Some samples contain long-term dependency information, while others only contain short-term information. In the following sections, we will go into the details of the proposed multi-term learning framework.

3 Method

The overview of MT-FGNE is shown in Figure 1, which consists of two main components and one plugin. The first component is FGN, which is an individual model to learn spatial and temporal dependencies. The other component is a multi-term ensemble learning framework, which constructs samples at different scales to enable the model to abstract both short-term and long-term dependencies. Considering that some sensor signals are generated when the equipment operates under various operation conditions, we designed a time series decomposition plugin to enhance the model's performance for such inputs.

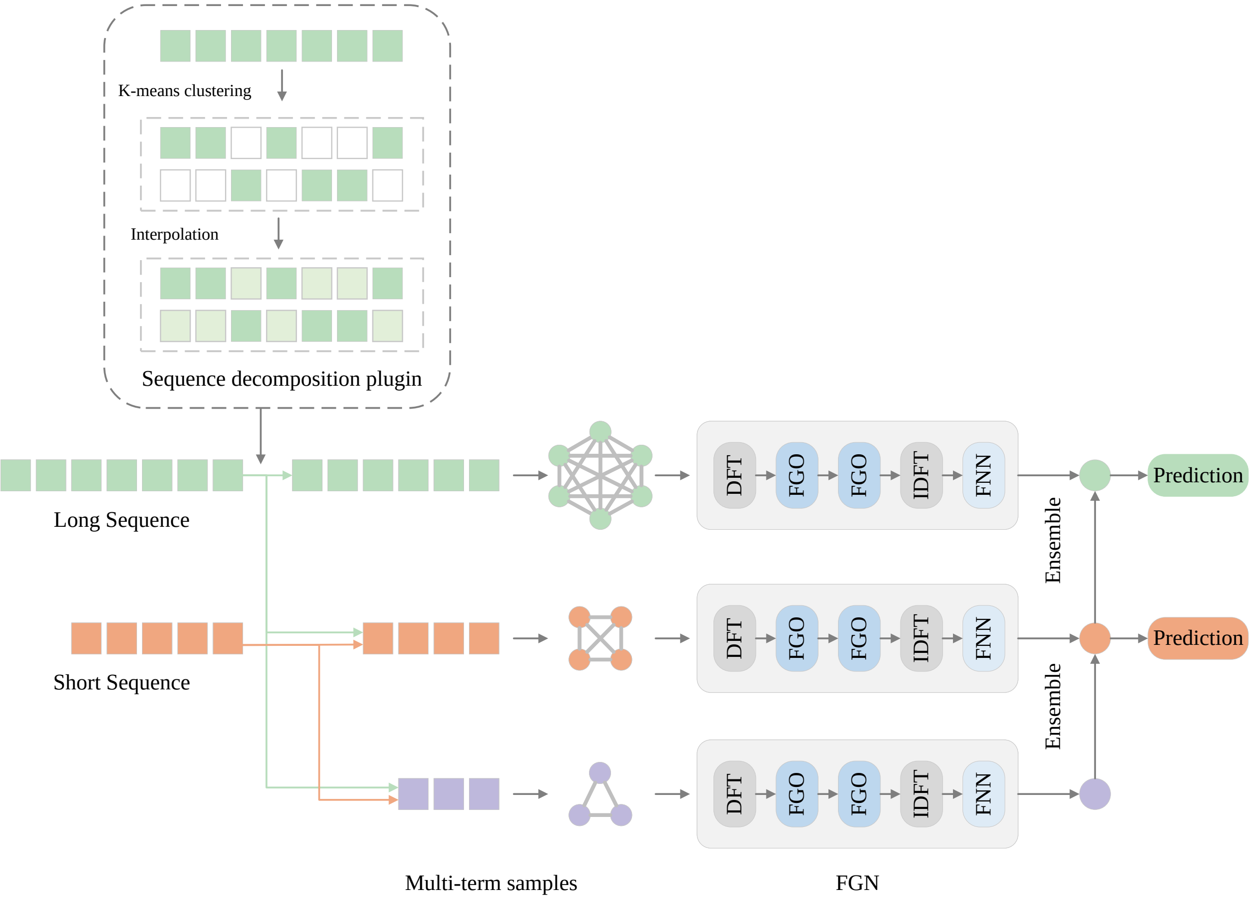

Figure 1: The overall framework of the proposed MT-FGNE. In MT-FGNE, a multi-term ensemble learning strategy is applied to construct samples at different scales to enable the model to abstract both short-term and long-term dependencies, and multiple FGNs are used to learn spatial and temporal dependencies within multi-term samples. Besides, a sequence decomposition plugin is designed to tackle sensor signals recorded under different operation conditions.

3.1 Preliminaries and motivations

Given multiple condition monitoring time series data $\left\{X^{(1)}, X^{(2)}, ..., X^{(M)}\right\}$, where $X^{(i)} = \left[\mathbf{x}_1^{(i)}, ..., \mathbf{x}_t^{(i)}, ..., \mathbf{x}_{L_i}^{(i)}\right] \in \mathbb{R}^{L_i \times N}$ represents the i-th time series with length L_i and feature dimension N, $\mathbf{x}_t^{(i)} \in \mathbb{R}^N$ represents the value of N features at timestamp t. We convert the raw time series into samples by applying the sliding time window approach, with the lookback window size T, each containing T observations at one time step as input features, and the corresponding output label $Y_t^{(i)}$. $X_t^{(i)} = \left[\mathbf{x}_{t-T+1}^{(i)}, \mathbf{x}_{t-T+2}^{(i)}, ..., \mathbf{x}_t^{(i)}\right] \in \mathbb{R}^{T \times N}$ denotes the input features of one sample at timestamps t. The RUL prediction task involves predicting the label $Y_t^{(i)}$ based on the input features $X_t^{(i)}$. When employing traditional sequential models to abstract the temporal information, the prediction process can be formulated by:

$$\hat{Y}_t^{(i)} := F_{\theta_t}(X_t^{(i)}) = F_{\theta_t}(\left[\mathbf{x}_{t-T+1}^{(i)}, \mathbf{x}_{t-T+2}^{(i)}, ..., \mathbf{x}_t^{(i)}\right]) \tag{1}$$

where $\hat{Y}_t^{(i)}$ are the forecasts matching the ground truth $Y_t^{(i)}$. The forecasting function is denoted as F_{θ_t} parameterized by θ_t. When using the ST-GNNs method, we first design the graphs or apply graph structure learning approaches to transform $\mathbf{x}_t^{(i)}$ into $\mathbf{g}_t^{(i)}$ at each timestep t, and then the RUL prediction can be expressed as the following formulation:

$$\hat{Y}_t^{(i)} := F_{\theta_t,\theta_g}(X_t^{(i)}) = F_{\theta_t,\theta_g}(\left[\mathbf{g}_{t-T+1}^{(i)}, \mathbf{g}_{t-T+2}^{(i)}, ..., \mathbf{g}_t^{(i)}\right]) \tag{2}$$

where the forecasting function is denoted as F_{θ_t,θ_g} parameterized by θ_t and θ_g, indicating ST-GNNs separately model spatial and temporal dependencies.

3.2 Fourier Graph Neural Networks

A recent study Yi et al. (2024) introduces FGN to address the oversight of potential spatiotemporal inter-dependencies that arise from modeling spatial and temporal dependencies separately in ST-GNNs. FGN aims to enhance learning efficiency by learning unified spatiotemporal dependencies. FGN no longer treats input samples as a sequence of graphs; instead, it regards them as one complete graph. Therefore, Equation 2 can be rewritten as:

$$\hat{Y}_t^{(i)} := FGN_{\theta_g}(X_t^{(i)}, A_t^{(i)}) \tag{3}$$

where $X_t^{(i)} \in \mathbb{R}^{(T\times N)\times 1}$, $A_t^{(i)} \in \{1\}^{(T\times N)\times(T\times N)}$ is the adjacency matrix of a complete graph, and θ_g are the parameters of the FGN. In FGN, we initially project the node features into a higher-dimensional space d to obtain node embeddings $X_t^{(i)} \in \mathbb{R}^{(T\times N)\times d}$, and perform a Discrete Fourier Transform (DFT) to transform node embeddings into the frequency domain and get $\mathcal{F}(X_t^{(i)}) \in \mathbb{C}^{(\left\lfloor \frac{(T\times N)}{2} \right\rfloor+1)\times d}$. Then we conduct recursive multiplications between $\mathcal{F}(X_t^{(i)})$ and Fourier Graph Operators (FGOs) in the Fourier space and make summations. Finally, we transform the node embeddings to the time domain using the Inverse Discrete Fourier Transform (IDFT), and utilize fully connected layers to map embeddings to labels, as illustrated in Figure 1. The detailed FGN procedure can be formulated as follows:

$$FGN_{\theta_g}(X_t^{(i)}, A_t^{(i)}) := \mathcal{F}^{-1}(\sum_{k=0}^{K} \sigma(\mathcal{F}(X_t^{(i)})S_{0:k} + b_k)), \quad S_{0:k} = \prod_{i=0}^{k} S_i. \tag{4}$$

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ stand for DFT and IDFT, respectively. $S_k \in \mathbb{C}^{d\times d}$ is the FGO in the k-th layer. σ is the activation function, and $b_k \in \mathbb{C}^d$ are the complex-valued bias parameters. By treating time series samples as complete graphs and performing transformations in the frequency domain, FGN can efficiently encode potential spatiotemporal inter-dependencies within sensor signal data while removing noise.

3.3 Multi-term ensemble learning framework

When employing the sliding time window approach to generate samples, one parameter that needs to be set is the lookback window size T. If only one fixed size of the time window exists, it cannot exceed the minimum length of the sequence; otherwise, some sequence data cannot be processed and predicted, so there is a constraint $T \leq \min_{i\in\{1,2,...,M\}} L_i$. This constraint often makes the chosen T too small and insufficient to contain long-term dependency information. Therefore, we propose a multi-term learning strategy. We apply a set of lookback windows instead of just one to generate samples:

$$T := \{T_{\min}, T_{\min} + 1 \times D, \ldots, T_{\min} + (C-1) \times D\} \tag{5}$$

where C indicates the lookback window numbers and D is the magnitude of the time window increment. $T_{\min}$ is the smallest time window we use, set to be smaller than the minimum length of the test sequences to ensure that the model can provide predictions for all test sequences. Then we gradually increase the time window to include more information in the samples. By using a set of different time window sizes, we generate multi-term samples, as shown in Figure 1.

In the training phase, we created multi-term training samples with multiple time window sizes. In the process of creating training samples, there may arise situations where the length of the training sequence is shorter than the set time window size. In such cases, we discarded the entire training sequence. Then we trained an FGN model for each scale of the training samples. This means the number of FGN models trained equaled the number of time windows we employed. Since the sizes of samples from different terms varied, training each model independently is advantageous for the model's learning process. After training, we obtain a group of trained FGNs: $\{FGN^{(1)}_{trained}, FGN^{(2)}_{trained}, \ldots, FGN^{(C)}_{trained}\}$. In existing methods, it is common practice to utilize the data within the last time window of the test sequence as the testing sample, inputting it into the trained model to obtain one prediction. In contrast, within our proposed framework, we generate multiple testing samples that correspond to the same label using different lookback window sizes. Some short sequences cannot generate a complete sample using larger time windows. This limitation becomes particularly evident during testing. Our strategy involves grouping the testing sequences based on the size of the time window used and conducting different tests for each group. As shown in Figure 1, longer sequences can generate multiple samples that are input into the corresponding trained models, resulting in multiple predictions. In contrast, shorter sequences can only utilize fewer or even only one model to obtain prediction results. Finally, we ensemble the multiple predictions to enhance prediction stability.

In summary, we adopted a multi-term ensemble learning approach to enhance the model's performance. The primary motivation behind this approach stems from our ability to generate more training samples and ensure they contain richer information from the same original time series data. Similarly, during testing, we generated multiple testing samples, thus improving the utilization of the testing data. Finally, by integrating the predictions obtained from multiple sources, we further enhanced the predictive performance and stability of the model.

3.4 Sequence decomposition under multiple operation conditions

Due to the equipment's varying operation conditions, the collected sensor signals often adhere to multiple distributions. A typical example of monitoring data with two operation conditions is illustrated in Figure 2. Sensor values frequently switch between two trajectories, exhibiting two distinct overall trends. With such overlapping signals from two different distributions, the risk of inaccurate identification of frequency components of the original signal arises, leading to aliasing errors Cao et al. (2022). Those with higher amplitudes might obscure signals with lower amplitudes, consequently diminishing the overall signal-to-noise ratio and potentially causing loss or distortion of signal information. This overlapping of signals with varying amplitudes adversely affects the learning of FGN in the frequency domain. To mitigate this issue, we employ a simple sequence decomposition method to denoise the input signals.

Our first step involves identifying operation conditions. If the condition monitoring data already includes records of operation conditions, we can analyze them directly. Alternatively, we extract information on operation conditions from sensor signals. We utilize the straightforward and efficient k-means Ahmed et al. (2020) method for clustering analysis of the raw sensor signals to determine which operation condition a given sensor signal belongs to at timestep t. Then we segment the sensor signals based on the identified operation conditions. This involves dividing the data into subsets, each corresponding to a specific operation condition, as shown in the following formulation.

$$X^{(i)} \xrightarrow{\text{k-means}} \{X^{(i,1)}, X^{(i,2)}, \ldots, X^{(i,k)}\} \tag{6}$$

where $X^{(i)}$ represents the i-th original time series, and $\{X^{(i,1)}, X^{(i,2)}, \ldots, X^{(i,k)}\}$ represent the time series resulting from the k-means clustering process, where k is the number of clusters obtained by the k-means algorithm. The newly generated multiple time series data are complementary to each other, with many missing values that require interpolation. While numerous novel interpolation methods have been proposed Oh et al. (2020), given the substantial number of data points requiring interpolation, we opt for the most straightforward method, equal-value interpolation. We use nearby values to fill in the missing values to enhance efficiency and avoid overfitting.

$$\begin{aligned} &\{X^{(i,1)}, X^{(i,2)}, \ldots, X^{(i,k)}\} \\ &\xrightarrow{\text{interpolation}} \{\hat{X}^{(i,1)}, \hat{X}^{(i,2)}, \ldots, \hat{X}^{(i,k)}\} \end{aligned} \tag{7}$$

Finally, we utilize a sliding time window approach to generate samples $\{\hat{X}_t^{(i,1)}, \hat{X}_t^{(i,2)}, \dots, \hat{X}_t^{(i,k)}\}$ for each operation condition sequence, inputting them into FGNs. The predicted mean under different operation conditions is then the final output.

$$\hat{Y}_t^{(i)} := \frac{1}{k} \sum_{l=1}^{k} FGN_{\theta_g}^{l}(\hat{X}_t^{(l,k)}, \hat{A}_t^{(l,k)}) \tag{8}$$

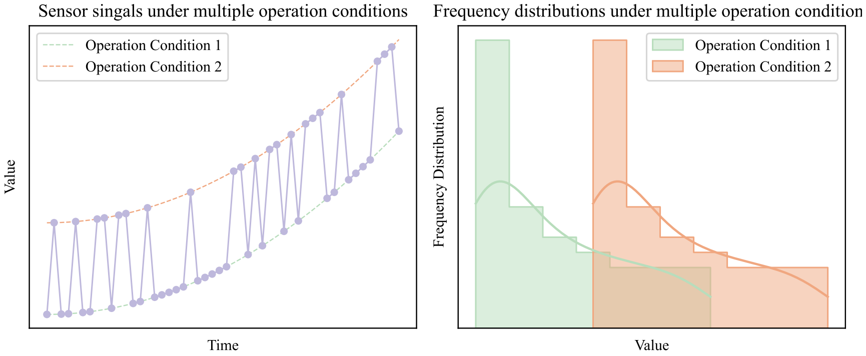

Figure 2: A typical example illustrates original sensor signals and their frequency distributions under multiple operation conditions.

4 Experiment and Results

In this section, we extensively evaluate the proposed MT-FGNE on a benchmark dataset.

4.1 Datasets description and analysis

The Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset is a widely used public dataset in the field of RUL prediction Xia et al. (2020). There are four subsets in this dataset. Each subset is divided into a training set and a test set. The training set contains multiple turbofan engine condition monitoring data from healthy operation to complete failure. The condition monitoring data in the test set ends before complete failure occurs. The goal is to forecast the engines' RUL in the test set. The characteristics of each subset are detailed in Table 1. Among the four data sets FD001-FD004, the engines in FD001 and FD003 operated under a single operation condition, while those in FD002 and FD004 operated under six different operation conditions, increasing the prediction complexity. Additionally, engines in FD001 and FD002 only have one fault mode, which is the High-Pressure Compressor (HPC) failure, while FD003 and FD004 hold two fault modes. Table1 also displays the minimum and maximum sequence lengths of the dataset, indicating significant variations in sequence lengths across different engines. Since training data captures engine operation up to the point of failure, the signal record is usually relatively long. Also, variations in initial engine states and failure processes lead to different sequence lengths for each engine. Existing models commonly use a single fixed lookback window to generate samples Kong et al. (2022); Wang et al. (2021); Chen & Zeng (2023). However, this lookback window size cannot exceed the minimum sequence length of the test engines; otherwise, the model cannot be applied to obtain predictions for all test engines. This relatively small lookback window size is inappropriate for test engines whose sensor data length is relatively large and may hinder the model's ability to learn long-term dependencies.

4.2 Implementation details

We maintain consistency in the data preprocessing settings with Kong et al. (2022); Wang et al. (2021) to ensure a fair comparison with existing models. We first normalize the 14 effective features selected from the

Table 1: Description of C-MAPSS turbofan engine dataset.

Subsets	Operation Condition	Fault Mode	Training units	Test units	Max length	Min length
FD001	1	HPC	100	100	362	31
FD002	6	HPC	260	259	378	21
FD003	1	HPC+Fan	100	100	525	38
FD004	6	HPC+Fan	249	248	543	19

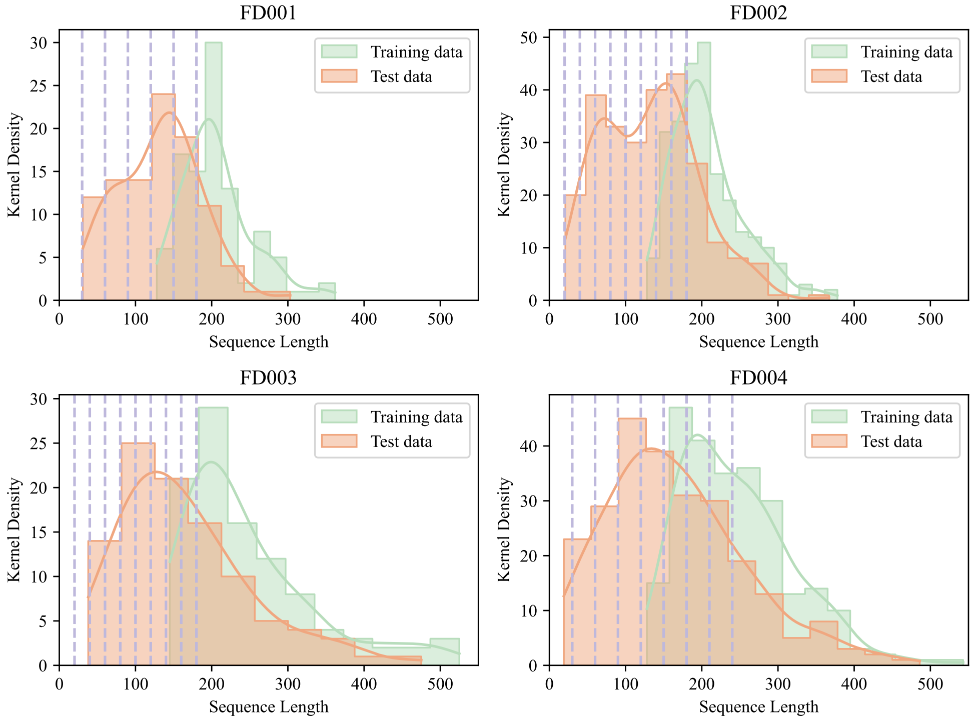

Figure 3: The training and test sequence length distribution of four subsets in the C-MAPSS dataset, multiple dashed lines parallel to the y-axis represent the various sizes of the lookback window we employed. These dashed lines partition the training and testing time series into multiple subgroups.

original 24 features. Then, we apply a piecewise function to rectify the training and test labels, ensuring that they do not exceed 125 to mitigate the possibility of overestimating RUL. Next, we apply the proposed multi-term learning approach by employing multiple lookback windows to generate various samples. For the four subsets, the lookback window sizes used are shown in Table 2. Unlike existing methods that use a single window size, which cannot exceed the minimum sequence length in the subset, resulting in short samples that fail to capture long-term dependencies in the time series, we adopt a set of lookback window sizes. We keep the first window size smaller than the minimum sequence length to ensure the applicability of the predictive model. Subsequently, we gradually increase the window size to generate longer samples, allowing the model to learn potential long-term dependencies. The employed lookback time windows and their comparison with the length of the time series are illustrated in Figure 3. We partition the training and testing sequences into subgroups using the multiple lookback windows we defined. We generate training and testing samples adaptively based on the length of sequences within each subgroup. Given the relatively longer sequences on the right side of each sub-figure, they could adopt all lookback windows on the left side of the sequences, while the reverse was not valid.

Table 2: Comparison of multiple time windows used in the proposed framework against existing single time windows.

Subsets	Min length	Single window size Kong et al. (2022)	Window sizes in MT-FGNE
FD001	31	30	30/60/90/120/150/180
FD002	21	20	20/40/60/80/100/120/140/160/180
FD003	38	30	30/60/90/120/150/180/210/240
FD004	19	15	18/40/62/84/106/128/150/172/194

Due to the operation of FD002 and FD004 data under six different operation conditions, directly applying FGN for frequency-domain learning may lead to a low signal-to-noise ratio. We select six features highly correlated with the labels ["s7", "s9", "s11", "s12", and "s13"] Huang et al. (2023) and cluster the data. After dividing it into six subsets, we employ a simple equal-value interpolation to fill in missing values. Subsequently, FGN is used to learn from each of the six subsequences separately. We set the number of FGO layers to three, which is deep enough for the RUL prediction task. The proposed model's training parameters were optimized using the RMSprop optimizer, with the following hyperparameters: 100 epochs, a learning rate of 0.001, and a batch size of 256. Two metrics are employed in the evaluation, including the Root Mean Square Error (RMSE) and a Score function Kong et al. (2022) as described in the following equation:

$$\text{Score}(v, \hat{v}_i) = \begin{cases} \sum_{i=1}^{M}(e^{-\frac{\hat{v}_i - v_i}{13}} - 1) & \text{if } \hat{v}_i < v_i \\ \sum_{i=1}^{M}(e^{\frac{\hat{v}_i - v_i}{10}} - 1) & \text{if } \hat{v}_i \geq v_i \end{cases} \tag{9}$$

The asymmetric Score function imposes a more significant penalty for overestimating RUL, as overestimating RUL leads to more severe consequences. Similarly to RMSE, a lower value of the score function indicates a better prediction performance.

4.3 Comparisons with state-of-the-art

This section compares our method with the most advanced RUL prediction techniques available. We mainly compare our method with ST-GNNs because of their outstanding performance on this task. Table 3 displays a comparison of different approaches on the C-MAPSS dataset. The current advanced approaches can be broadly categorized into two groups. The first group consists of sequence models, with the most popular being Transformer-based models. The second group comprises ST-GNNs, which excel in predictive performance by capturing spatial information, generally outperforming sequence models. Our approach differs significantly from existing methods. First, FGN conducts learning in the frequency domain space instead of the time domain. Secondly, we transform samples into a graph rather than a sequence of graphs. Finally, we adopt a multi-term learning approach to enhance the model's learning of long-term dependencies within samples.

We compare our method with several best models in the literature. The values reported in Table 3 are taken directly from the papers as their implementations are not publicly available. One immediate observation on the performances of these baseline models is that there is no single best model that is capable of outperforming others on all four datasets, demonstrating the difficulty of the prediction task and the diversity shown in the datasets. In comparison to these baselines, our proposed MT-FGNE demonstrates superior performance on the first three datasets, FD001, FD002, and FD003. It shows a 13.6%, 2.3%, and 5.3% improvement compared to the second-best baseline on these three datasets, respectively. On FD004, our model is ranked second among eighteen models in terms of RMSE, with only small gaps to the best-performing models. The results indicate that even under various operation conditions and relatively low signal-to-noise ratios in original sensor signals, FGN can effectively learn dependencies after proper sequence decomposition and processing. Compared to traditional ST-GNNs, our method does not require learning graph structures or separately learning spatial and temporal information. Moreover, the Fourier Graph Operator is significantly less computationally expensive than the message-passing operators of GNNs. We follow STFA Kong et al. (2022) and create an ST-GNN. The training time of ST-GNN and FGN on FD001 is 2518s and 729s, and the trainable parameters of ST-GNN and FGN are 183,905 and 74,064, respectively, showing that FGN has a lower computational burden. Overall, our method consistently delivers outstanding performance across different scenarios, demonstrating its ability to provide significant contributions to RUL prediction.

Table 3: Compare the RMSE and Score values of MT-FGNE with other advanced sequence models and ST-GNN methods for the C-MAPSS dataset (bold: best; underline: runner-up).

Models	FD001		FD002		FD003		FD004		Average	
	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
DA-Transformer Liu et al. (2022)	12.25	198	17.08	1575	13.39	290	19.86	1741	15.65	951.00
BiGRU-TSAM Zhang et al. (2022a)	12.56	213	18.94	2264	12.45	233	20.47	3610	16.11	1580.00
MSIDSN Zhao et al. (2023)	11.74	206	18.26	2047	12.04	196	22.48	2911	16.13	1340.00
EAPN Zhang et al. (2023)	12.11	245	15.68	1127	12.52	267	18.12	2051	14.61	922.50
Crossformer Wang et al. (2023b)	12.11	216	14.16	837	12.32	260	14.81	956	13.35	567.25
HAGCN Li et al. (2021)	11.93	222	15.05	1144	11.53	240	15.74	1219	13.56	706.25
STGCN Wang et al. (2021)	14.55	402	14.58	943	13.06	394	14.60	1065	14.20	701.00
STFA Kong et al. (2022)	11.35	194	19.17	2493	11.64	225	21.41	2760	15.89	1418.00
DAST Zhang et al. (2022b)	11.43	203	15.25	925	11.32	**155**	18.36	1491	14.09	693.50
GGCN Wang et al. (2022)	11.82	187	17.24	1494	12.21	245	17.36	1372	14.66	824.50
ConvGAT Chen & Zeng (2023)	11.34	197	14.12	772	10.97	235	15.51	1231	12.99	608.75
CDSG Wang et al. (2023a)	11.26	188	18.13	1740	12.03	218	19.73	2332	15.29	1119.50
DCFA Gao et al. (2023)	11.74	190	16.81	1076	10.71	198	17.77	1571	14.26	758.75
LOGO Wang et al. (2023b)	12.13	226	13.54	832	12.18	261	**14.29**	**944**	13.04	565.75
NSD-TGTN Gao et al. (2024)	12.13	226	15.87	1477	12.01	220	16.64	1493	14.16	854.00
DVGTformer Wang et al. (2024)	11.33	180	14.28	797	11.89	255	15.50	1108	13.25	585.00
THGNN Wen et al. (2024)	13.15	285	13.84	806	12.61	255	14.65	1166	13.56	628.00
MT-FGNE	**9.73**	**152**	**13.23**	**694**	**10.14**	178	14.40	958	**11.88**	**495.50**

4.4 Ablation studies

To demonstrate the effectiveness of the proposed framework, we conducted an ablation study comparing MT-FGNE with its variants. We primarily evaluated two components of MT-FGNE: the multi-term ensemble learning (MTE) framework, and the sequence decomposition (SD) plugin.

4.4.1 Multi-term ensemble ablations

The first variant is the single FGN model. Like existing methods, individual FGN adopts smaller, fixed time windows to generate samples at a certain scale, resulting in unsatisfactory predictive performance. As shown in Table 4, compared to individual FGN, our MT-FGNE reduced 18.0% in prediction error on the FD001 dataset. Similar effects are observed across all four datasets, demonstrating the high applicability of the proposed framework. As analyzed earlier, the multi-term ensemble learning framework primarily relies on generating longer samples to enhance the model's ability to learn long-term dependencies. The prerequisite for performance improvement is that the adopted model possesses strong sequence modeling capabilities. For instance, MT-FGNE w/o SD applied multi-term learning, yet its performance on FD002 did not show significant enhancement. It is attributed to the sensitivity of the adopted model to noise in sensor signals, leading to poor learning capabilities. Merely increasing the time window does not contribute to resolving the noise issue in this situation.

The proposed multi-term ensemble learning framework is not limited to a specific individual model for learning and theoretically can be applied to all existing models to further enhance performance. We employed a simple sequence modeling tool, CNN, as the individual model. Following the adoption of multi-term learning, its predictive performance saw significant improvement. Particularly noteworthy is its performance on the FD002 and FD004 datasets, where the prediction error surpassed that of MT-FGNE and even outperformed the current state-of-the-art model by 16.2%. This result suggests that employing smaller, fixed time window settings hinders the model's ability to capture long-range dependencies in the data, thereby limiting the performance of most existing models. Figure 4 illustrates the performance of individual FGN models and ensemble results on the last four test subsets of the FD001 dataset. In subset 3, the test sequence length is insufficient to apply all FGN models, while in subset 6, the sequence length exceeds the maximum time window, set at 180, thus allowing the application of all trained FGN models for prediction. On each test subset, FGN models trained with samples generated from larger time windows demonstrate superior predictive performance, as they are better at capturing long-term dependencies. In most cases, the ensemble predictions further enhance accuracy and significantly reduce the variance of prediction errors, indicating improved stability of the predictive models.

Table 4: Ablation study on the C-MAPSS dataset.

Variants	FGN	MTE	SD	FD001		FD002		FD003		FD004	
				RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
CNN	✗	✗	✗	12.71	241	15.74	1504	11.47	185	17.81	1809
MT-CNNE	✗	✓	✗	11.76	248	**11.83**	**537**	10.77	240	**13.76**	**853**
FGN	✓	✗	✗	11.87	194	20.87	2221	12.64	208	27.22	4777
MT-FGNE w/o SD	✓	✓	✗	-	-	19.21	1563	-	-	23.38	2781
MT-FGNE w/o MTE	✓	✗	✓	-	-	16.39	1411	-	-	18.37	2168
MT-FGNE	✓	✓	✓	**9.73**	**152**	13.23	694	**10.14**	**178**	14.40	958

MTE: Multi-Term Ensemble, SD: Sequence Decomposition.

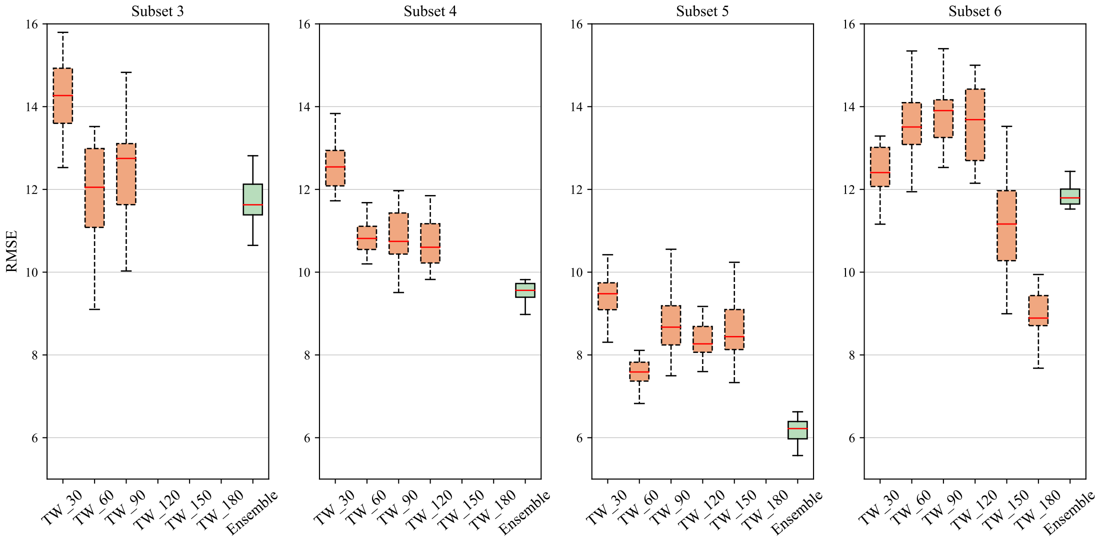

Figure 4: The individual FGN models' performance and ensemble results on the last four test subsets in FD001.

4.4.2 Sequence decomposition ablations

The FD002 and FD004 datasets were generated under six different operation conditions. Our study primarily focuses on these two datasets to validate the effectiveness of sequence decomposition. As shown in Table

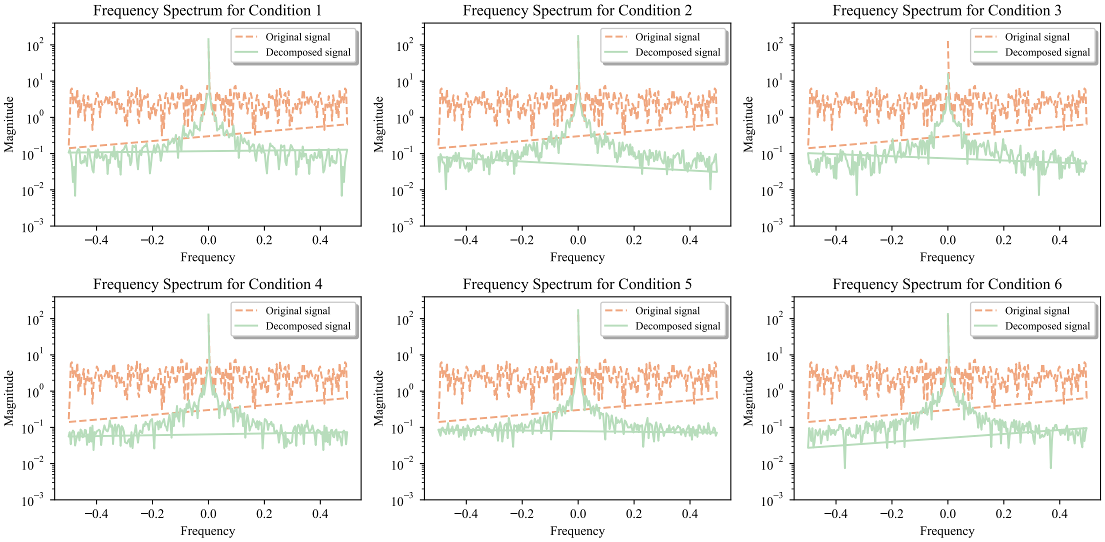

Figure 5: The frequency spectrum comparison of raw sensor signals and signals after decomposition in FD002.

4, although the individual FGN w/o SD method exhibits decent predictive performance on FD001 and FD003, its predictive errors on FD002 and FD004 are quite significant. By comparing FGN with MT-FGNE w/o SD, we can find that the prediction performance of MT-FGNE w/o SD on FD002 is not significantly improved even with the MTE strategy. It indicates that even when the samples contain more information on long-term dependencies, the FGN model fails to learn from them effectively, diminishing the value of MTE. In contrast to FGN, the MT-FGNE w/o MTE model incorporates the SD plugin, resulting in a substantial improvement in predictive performance on FD002 and FD004. Figure 5 illustrates the frequency spectrum of an FD002 sample's original signal and the signal after denoising with SD. We observe a more uniform energy distribution in the original signal with frequent random fluctuations in frequency and no clear frequency range with high-amplitude signals. After denoising with SD, there is a significant increase in amplitude in the low-frequency region of the frequency spectrum, indicating an enhancement in the signal-to-noise ratio. Finally, a significant enhancement in predictive performance is observed when combining SD for signal denoising, followed by the MTE module. This phenomenon underscores the necessity of applying SD for denoising signals, particularly for datasets with multiple operation conditions.

5 Conclusion

This paper investigates the RUL prediction problem. Unlike existing deep learning models with fixed-size input samples, we consider the diversity in the time series lengths and propose a model capable of learning multi-term dependencies. We adaptively employ multiple lookback windows based on the time series length to generate multiple samples with the same label. Subsequently, these samples are transformed into complete graphs, and FGN models are employed to learn both spatial and temporal dependencies across multiple terms. The predictions from FGN models are then integrated to obtain the final prediction. This approach enhances the model's capability to learn long-range temporal dependencies in long sequences and further improves prediction performance by integrating predictions from different terms. For condition monitoring data generated under various operation conditions, we cluster the data based on operation conditions and then interpolate missing values, learn spectral features through FGN after DFT on data from different operation conditions, and finally average the prediction results. Extensive experimental results demonstrate that our proposed MT-FGNE model achieves state-of-the-art predictive performance on the C-MAPSS turbofan engine dataset.

References

Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8):1295, 2020.

Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 354–370. Springer, 2016.

Jiahui Cao, Shaohua Tian, Zhibo Yang, Guangrong Teng, Haoqi Li, Ruochen Jin, Ruqiang Yan, and Xuefeng Chen. Blade tip timing signal filtering method based on sampling aliasing frequency map. *IEEE Transactions on Instrumentation and Measurement*, 71:1–12, 2022.

Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 357–366, 2021a.

Ling Chen, Donghui Chen, Zongjiang Shang, Binqing Wu, Cen Zheng, Bo Wen, and Wei Zhang. Multi-scale adaptive graph neural network for multivariate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

Wei Chen and Ke Shi. Multi-scale attention convolutional neural network for time series classification. *Neural Networks*, 136:126–140, 2021.

Xiao Chen and Ming Zeng. Convolution-graph attention network with sensor embeddings for remaining useful life prediction of turbofan engines. *IEEE Sensors Journal*, 2023.

Xiaohan Chen, Beike Zhang, and Dong Gao. Bearing fault diagnosis base on multi-scale cnn and lstm model. *Journal of Intelligent Manufacturing*, 32(4):971–987, 2021b.

Zhicheng Cui, Wenlin Chen, and Yixin Chen. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995*, 2016.

Carlos Ferreira and Gil Gonçalves. Remaining useful life prediction and challenges: A literature review on the use of machine learning methods. *Journal of Manufacturing Systems*, 63:550–562, 2022.

Hui Gao, Yibin Li, Ying Zhao, and Yan Song. Dual channel feature-attention-based approach for rul prediction considering the spatiotemporal difference of multisensor data. *IEEE Sensors Journal*, 2023.

Zhan Gao, Weixiong Jiang, Jun Wu, Tianjiao Dai, and Haiping Zhu. Nonlinear slow-varying dynamics-assisted temporal graph transformer network for remaining useful life prediction. *Reliability Engineering & System Safety*, 248:110162, 2024.

Lin Huang, Xin Pan, Yajie Liu, and Li Gong. An unsupervised machine learning approach for monitoring data fusion and health indicator construction. *Sensors*, 23(16):7239, 2023.

Li Jiang, Tianao Zhang, Wei Lei, Kejia Zhuang, and Yibing Li. A new convolutional dual-channel transformer network with time window concatenation for remaining useful life prediction of rolling bearings. *Advanced Engineering Informatics*, 56:101966, 2023.

Guangyin Jin, Yuxuan Liang, Yuchen Fang, Zezhi Shao, Jincai Huang, Junbo Zhang, and Yu Zheng. Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

Ziqian Kong, Xiaohang Jin, Zhengguo Xu, and Bin Zhang. Spatio-temporal fusion attention: A novel approach for remaining useful life prediction based on graph neural network. *IEEE Transactions on Instrumentation and Measurement*, 71:1–12, 2022.

Tianfu Li, Zhibin Zhao, Chuang Sun, Ruqiang Yan, and Xuefeng Chen. Hierarchical attention graph convolutional network to fuse multi-sensor signals for remaining useful life prediction. *Reliability Engineering & System Safety*, 215:107878, 2021.

Xinyao Li, Jingjing Li, Lin Zuo, Lei Zhu, and Heng Tao Shen. Domain adaptive remaining useful life prediction with transformer. *IEEE Transactions on Instrumentation and Measurement*, 71:1–13, 2022.

Lu Liu, Xiao Song, and Zhetao Zhou. Aircraft engine remaining useful life estimation via a double attention-based data-driven architecture. *Reliability Engineering & System Safety*, 221:108330, 2022.

Cheolhwan Oh, Seungmin Han, and Jongpil Jeong. Time-series data augmentation based on interpolation. *Procedia Computer Science*, 175:64–71, 2020.

Lei Ren, Jiabao Dong, Xiaokang Wang, Zihao Meng, Li Zhao, and M Jamal Deen. A data-driven auto-cnn-lstm prediction model for lithium-ion battery remaining useful life. *IEEE Transactions on Industrial Informatics*, 17(5):3478–3487, 2020.

Zunya Shi and Abdallah Chehade. A dual-lstm framework combining change point detection and remaining useful life prediction. *Reliability Engineering & System Safety*, 205:107257, 2021.

Hongfei Wang, Zhuo Zhang, Xiang Li, Xinyang Deng, and Wen Jiang. Comprehensive dynamic structure graph neural network for aero-engine remaining useful life prediction. *IEEE Transactions on Instrumentation and Measurement*, 2023a.

Lei Wang, Hongrui Cao, Hao Xu, and Haichen Liu. A gated graph convolutional network with multi-sensor signals for remaining useful life prediction. *Knowledge-Based Systems*, 252:109340, 2022.

Lei Wang, Hongrui Cao, Zhisheng Ye, Hao Xu, and Jiaxiang Yan. Dvgtformer: A dual-view graph transformer to fuse multi-sensor signals for remaining useful life prediction. *Mechanical Systems and Signal Processing*, 207:110935, 2024.

Mengni Wang, Yuanxiang Li, Yuxuan Zhang, and Lei Jia. Spatio-temporal graph convolutional neural network for remaining useful life estimation of aircraft engines. *Aerospace Systems*, 4(1):29–36, 2021.

Yucheng Wang, Min Wu, Ruibing Jin, Xiaoli Li, Lihua Xie, and Zhenghua Chen. Local–global correlation fusion-based graph neural network for remaining useful life prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 2023b.

Zhihao Wen, Yuan Fang, Pengcheng Wei, Fayao Liu, Zhenghua Chen, and Min Wu. Temporal and heterogeneous graph neural network for remaining useful life prediction. *arXiv preprint arXiv:2405.04336*, 2024.

Ji-Yan Wu, Min Wu, Zhenghua Chen, Xiao-Li Li, and Ruqiang Yan. Degradation-aware remaining useful life prediction with lstm autoencoder. *IEEE Transactions on Instrumentation and Measurement*, 70:1–10, 2021.

Tangbin Xia, Ya Song, Yu Zheng, Ershun Pan, and Lifeng Xi. An ensemble framework based on convolutional bi-directional lstm with multiple time windows for remaining useful life estimation. *Computers in Industry*, 115:103182, 2020.

Xingwei Xu, Xiang Li, Weiwei Ming, and Ming Chen. A novel multi-scale cnn and attention mechanism method with multi-sensor signal for remaining useful life prediction. *Computers & Industrial Engineering*, 169:108204, 2022.

Boyuan Yang, Ruonan Liu, and Enrico Zio. Remaining useful life prediction based on a double-convolutional neural network architecture. *IEEE Transactions on Industrial Electronics*, 66(12):9521–9530, 2019.

Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. *Advances in Neural Information Processing Systems*, 36, 2024.

Jiusi Zhang, Yuchen Jiang, Shimeng Wu, Xiang Li, Hao Luo, and Shen Yin. Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism. *Reliability Engineering & System Safety*, 221:108297, 2022a.

Xiong Zhang, Yunfei Guo, Hong Shangguan, Ranran Li, Xiaojia Wu, and Anhong Wang. Predicting remaining useful life of a machine based on embedded attention parallel networks. *Mechanical Systems and Signal Processing*, 192:110221, 2023.

Zhizheng Zhang, Wen Song, and Qiqiang Li. Dual-aspect self-attention based on transformer for remaining useful life prediction. *IEEE Transactions on Instrumentation and Measurement*, 71:1–11, 2022b.

Ke Zhao, Zhen Jia, Feng Jia, and Haidong Shao. Multi-scale integrated deep self-attention network for predicting remaining useful life of aero-engine. *Engineering Applications of Artificial Intelligence*, 120: 105860, 2023.

Yexu Zhou, Michael Hefenbrock, Yiran Huang, Till Riedel, and Michael Beigl. Automatic remaining useful life estimation framework with embedded convolutional lstm as the backbone. In *Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part IV*, pp. 461–477. Springer, 2021.

Brahim Zraibi, Chafik Okar, Hicham Chaoui, and Mohamed Mansouri. Remaining useful life assessment for lithium-ion batteries using cnn-lstm-dnn hybrid method. *IEEE Transactions on Vehicular Technology*, 70 (5):4252–4261, 2021.